SONY P11850Q0 4

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２０００年 ７月２４日

出願番号
Application Number: 特願２０００－２２２１２５

出願人
Applicant(s): ソニー株式会社

２００１年 ５月１１日

特許庁長官
Commissioner,
Japan Patent Office 及川耕造

出証番号 出証特２００１－３０３８６４２

【書類名】　特許願
【整理番号】　00006009
【提出日】　平成12年 7月24日
【あて先】　特許庁長官殿
【国際特許分類】　G06F 19/00
【発明の名称】　データ処理装置およびデータ処理方法、並びにプログラム提供媒体
【請求項の数】　15
【発明者】
【住所又は居所】　東京都品川区北品川６丁目７番３５号　ソニー株式会社内
【氏名】　岡上　拓己
【特許出願人】
【識別番号】　000002185
【氏名又は名称】　ソニー株式会社
【代表者】　出井　伸之
【代理人】
【識別番号】　100101801
【弁理士】
【氏名又は名称】　山田　英治
【電話番号】　03-5541-7577
【選任した代理人】
【識別番号】　100093241
【弁理士】
【氏名又は名称】　宮田　正昭
【電話番号】　03-5541-7577
【選任した代理人】
【識別番号】　100086531
【弁理士】

【氏名又は名称】　澤田　俊夫

【電話番号】　03-5541-7577

【手数料の表示】

【予納台帳番号】　062721

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9904833

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　データ処理装置およびデータ処理方法、並びにプログラム提供媒体

【特許請求の範囲】

【請求項１】

記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行するデータ処理装置であり、

前記検証値は、コンテンツのカテゴリ毎に独立した検証値として生成して格納する構成を有することを特徴とするデータ処理装置。

【請求項２】

前記データ処理装置は、

コンテンツ利用に際して、利用対象コンテンツ構成データに基づいて検証値を算出し、該算出検証値と、予め格納された検証値との比較処理を実行し、一致する場合にのみコンテンツ利用を可能とした構成を有することを特徴とする請求項１に記載のデータ処理装置。

【請求項３】

前記記憶装置は、複数のディレクトリの各々に異なるカテゴリのコンテンツデータを格納する構成を有し、

前記検証値は、前記複数のディレクトリの各々を単位とするコンテンツの集合に対して生成される値であることを特徴とする請求項１に記載のデータ処理装置。

【請求項４】

前記記憶装置は、フラッシュメモリであり、

前記カテゴリ毎の検証値は、フラッシュメモリの使用禁止ブロックとして設定された領域に格納される構成を有することを特徴とする請求項１に記載のデータ処理装置。

【請求項５】

前記カテゴリは、コンテンツの種類に基づいて設定された構成であり、

コンテンツの種類別に独立した検証値を設定し、格納する構成を有することを特徴とする請求項１に記載のデータ処理装置。

【請求項６】

前記カテゴリは、コンテンツの暗号処理鍵として設定されるコンテンツキーＫｃｏｎを暗号化して提供する有効化キーブロック（ＥＫＢ）の管理エンティテイに基づいて設定された構成であり、

有効化キーブロック（ＥＫＢ）の管理エンティテイ別に独立した検証値を設定し、格納する構成を有することを特徴とする請求項１に記載のデータ処理装置。

【請求項７】

前記検証値は、検証対象となるコンテンツおよびヘッダ情報等のコンテンツ関連データを構成する部分データをメッセージとしてＤＥＳ暗号処理によって生成されるメッセージ認証符号（ＭＡＣ）に基づいて生成されるデータであることを特徴とする請求項１に記載のデータ処理装置。

【請求項８】

記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行するデータ処理方法であり、

前記検証値を、コンテンツのカテゴリ毎に独立した検証値として生成して格納することを特徴とするデータ処理方法。

【請求項９】

前記データ処理方法において、さらに、

コンテンツ利用に際して、利用対象コンテンツ構成データに基づいて検証値を算出し、該算出検証値と、予め格納された検証値との比較処理を実行し、一致する場合にのみコンテンツ利用を行なうことを特徴とする請求項８に記載のデータ処理方法。

【請求項１０】

前記記憶装置は、複数のディレクトリの各々に異なるカテゴリのコンテンツデータを格納する構成を有し、

前記検証値は、前記複数のディレクトリの各々を単位とするコンテンツの集合

に対して生成することを特徴とする請求項８に記載のデータ処理方法。

【請求項１１】

前記記憶装置は、フラッシュメモリであり、

前記カテゴリ毎の検証値は、フラッシュメモリの使用禁止ブロックとして設定された領域に格納することを特徴とする請求項８に記載のデータ処理方法。

【請求項１２】

前記カテゴリは、コンテンツの種類に基づいて設定され、

コンテンツの種類別に独立した検証値を設定し、格納することを特徴とする請求項８に記載のデータ処理方法。

【請求項１３】

前記カテゴリは、コンテンツの暗号処理鍵として設定されるコンテンツキーＫｃｏｎを暗号化して提供する有効化キーブロック（ＥＫＢ）の管理エンティテイに基づいて設定され、

有効化キーブロック（ＥＫＢ）の管理エンティテイ別に独立した検証値を設定し、格納することを特徴とする請求項８に記載のデータ処理方法。

【請求項１４】

前記検証値は、検証対象となるコンテンツおよびヘッダ情報等のコンテンツ関連データを構成する部分データをメッセージとしてＤＥＳ暗号処理によって生成されるメッセージ認証符号（ＭＡＣ）に基づいて生成されるデータであることを特徴とする請求項８に記載のデータ処理方法。

【請求項１５】

記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行するデータ処理をコンピュータ・システム上で実行せしめるコンピュータ・プログラムを提供するプログラム提供媒体であって、前記コンピュータ・プログラムは、

検証値を、コンテンツのカテゴリ毎に独立した検証値として生成して格納するステップを有することを特徴とするプログラム提供媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、データ処理装置およびデータ処理方法、並びにプログラム提供媒体に関する。特に、記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行する構成において、検証値をコンテンツのカテゴリ毎に独立した検証値として生成して格納する構成とし、コンテンツ改竄チェック処理の効率化を実現したデータ処理装置およびデータ処理方法、並びにプログラム提供媒体に関する。

【０００２】

【従来の技術】

昨今、音楽データ、ゲームプログラム、画像データ等、様々なソフトウエアデータ（以下、これらをコンテンツ（Content）と呼ぶ）を、インターネット等のネットワーク、あるいは、メモリカード、ＤＶＤ、ＣＤ等の流通可能な記憶媒体を介して流通させるコンテンツ流通が盛んになってきている。これらの流通コンテンツは、ユーザの所有するＰＣ（Personal Computer）、再生専用器、あるいはゲーム機器におけるコンテンツデータの受信、あるいはメモリカード、ＣＤ、ＤＶＤ等の記憶媒体の装着により、コンテンツ再生処理が実行されたり、あるいは外部からの入力コンテンツを再生器、ＰＣ等に内蔵の記録デバイス、例えばメモリカード、ハードディスク等に格納し、再度、格納媒体から再生する等の方法により利用される。

【０００３】

再生装置、ゲーム機器、ＰＣ等の情報機器には、流通コンテンツをネットワークから受信するため、あるいはＤＶＤ、ＣＤ等にアクセスするためのインタフェースを有し、さらにコンテンツの再生に必要となる制御手段、プログラム、データのメモリ領域として使用されるＲＡＭ、ＲＯＭ等を有する。

【０００４】

音楽データ、画像データ、あるいはプログラム等の様々なコンテンツは、再生機器として利用される再生装置、ゲーム機器、ＰＣ等の情報機器本体からのユーザ指示、あるいは接続された入力手段を介したユーザの指示により、例えば内蔵、あるいは着脱自在の記憶媒体から呼び出され、情報機器本体、あるいは接続さ

れたディスプレイ、スピーカ等を通じて再生される。

【０００５】

ゲームプログラム、音楽データ、画像データ等、多くのソフトウエア・コンテンツは、一般的にその作成者、販売者に頒布権等が保有されている。従って、これらのコンテンツの配布に際しては、一定の利用制限、すなわち、正規なユーザに対してのみ、ソフトウエアの使用を許諾し、許可のない複製等が行われないようにする、すなわちセキュリティを考慮した構成をとるのが一般的となっている。

【０００６】

ユーザに対する利用制限を実現する１つの手法が、配布コンテンツの暗号化処理である。すなわち、例えばインターネット等を介して暗号化された音声データ、画像データ、ゲームプログラム等の各種コンテンツを配布するとともに、正規ユーザであると確認された者に対してのみ、配布された暗号化コンテンツを復号する手段、すなわち復号鍵を付与する構成である。

【０００７】

暗号化データは、所定の手続きによる復号化処理によって利用可能な復号データ（平文）に戻すことができる。このような情報の暗号化処理に暗号化鍵を用い、復号化処理に復号化鍵を用いるデータ暗号化、復号化方法は従来からよく知られている。

【０００８】

暗号化鍵と復号化鍵を用いるデータ暗号化・復号化方法の態様には様々な種類あるが、その１つの例としていわゆる共通鍵暗号化方式と呼ばれている方式がある。共通鍵暗号化方式は、データの暗号化処理に用いる暗号化鍵とデータの復号化に用いる復号化鍵を共通のものとして、正規のユーザにこれら暗号化処理、復号化に用いる共通鍵を付与して、鍵を持たない不正ユーザによるデータアクセスを排除するものである。この方式の代表的な方式にＤＥＳ（データ暗号標準：Deta encryption standard）がある。

【０００９】

上述の暗号化処理、復号化に用いられる暗号化鍵、復号化鍵は、例えばあるパ

スワード等に基づいてハッシュ関数等の一方向性関数を適用して得ることができる。一方向性関数とは、その出力から逆に入力を求めるのは非常に困難となる関数である。例えばユーザが決めたパスワードを入力として一方向性関数を適用して、その出力に基づいて暗号化鍵、復号化鍵を生成するものである。このようにして得られた暗号化鍵、復号化鍵から、逆にそのオリジナルのデータであるパスワードを求めることは実質上不可能となる。

【００１０】

また、暗号化するときに使用する暗号化鍵による処理と、復号するときに使用する復号化鍵の処理とを異なるアルゴリズムとした方式がいわゆる公開鍵暗号化方式と呼ばれる方式である。公開鍵暗号化方式は、不特定のユーザが使用可能な公開鍵を使用する方法であり、特定個人に対する暗号化文書を、その特定個人が発行した公開鍵を用いて暗号化処理を行なう。公開鍵によって暗号化された文書は、その暗号化処理に使用された公開鍵に対応する秘密鍵によってのみ復号処理が可能となる。秘密鍵は、公開鍵を発行した個人のみが所有するので、その公開鍵によって暗号化された文書は秘密鍵を持つ個人のみが復号することができる。公開鍵暗号化方式の代表的なものにはＲＳＡ（Rivest-Shamir-Adleman）暗号がある。このような暗号化方式を利用することにより、暗号化コンテンツを正規ユーザに対してのみ復号可能とするシステムが可能となる。

【００１１】

【発明が解決しようとする課題】

コンテンツデータの正当性、すなわちデータが改竄されていないことをチェックするために検証用のチェック値を正規のコンテンツデータに基づいて生成して予めメモリに格納し、コンテンツ利用時に検証対象のデータに基づいて生成したチェック値と格納チェック値とを照合処理することによって、データ検証を行なう方法が従来から行なわれている。

【００１２】

しかしながら、メモリに格納するコンテンツ数が増大すると、検証用のチェック値を正規のコンテンツデータに基づいて生成し、格納し管理することが困難となる。特に、昨今フラッシュメモリを使用したメモリカード等の容量の大きい媒

体においては、音楽データ、画像データ、プログラムデータ等、様々なカテゴリのコンテンツデータがメモリに格納されることとなる。このような環境においては、コンテンツ。チェック値の生成処理、格納処理、改竄チェック処理の管理は困難となる。格納データ全体に対するチェック値を生成すると、チェック対象となったデータ全体に対するチェック値生成処理を実行することが必要となる。例えばＤＥＳ－ＣＢＣモードにおいて生成されるメッセージ認証符号（ＭＡＣ）により、チェック値ＩＣＶを求める手法を行なう場合、データ全体に対するＤＥＳ－ＣＢＣの処理を実行することが必要となる。この計算量は、データ長が長くなるにつれ増大することとなり、処理効率の点で問題がある。

【００１３】

本発明は、このような従来技術の問題点を解決するものであり、データ正当性の確認処理を効率的に実行し、コンテンツデータの検証処理の効率化、さらに検証後の記録デバイスに対するダウンロード処理、あるいは検証後の再生処理等を効率的に実行することを可能とするデータ処理装置およびデータ処理方法、並びにプログラム提供媒体を提供することを目的とする。

【００１４】

【課題を解決するための手段】

本発明の第１の側面は、

記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行するデータ処理装置であり、

前記検証値は、コンテンツのカテゴリ毎に独立した検証値として生成して格納する構成を有することを特徴とするデータ処理装置にある。

【００１５】

さらに、本発明のデータ処理装置の一実施態様において、前記データ処理装置は、コンテンツ利用に際して、利用対象コンテンツ構成データに基づいて検証値を算出し、該算出検証値と、予め格納された検証値との比較処理を実行し、一致する場合にのみコンテンツ利用を可能とした構成を有することを特徴とする。

【００１６】

さらに、本発明のデータ処理装置の一実施態様において、前記記憶装置は、複数のディレクトリの各々に異なるカテゴリのコンテンツデータを格納する構成を有し、前記検証値は、前記複数のディレクトリの各々を単位とするコンテンツの集合に対して生成される値であることを特徴とする。

【００１７】

さらに、本発明のデータ処理装置の一実施態様において、前記記憶装置は、フラッシュメモリであり、前記カテゴリ毎の検証値は、フラッシュメモリの使用禁止ブロックとして設定された領域に格納される構成を有することを特徴とする。

【００１８】

さらに、本発明のデータ処理装置の一実施態様において、前記カテゴリは、コンテンツの種類に基づいて設定された構成であり、コンテンツの種類別に独立した検証値を設定し、格納する構成を有することを特徴とする。

【００１９】

さらに、本発明のデータ処理装置の一実施態様において、前記カテゴリは、コンテンツの暗号処理鍵として設定されるコンテンツキーＫｃｏｎを暗号化して提供する有効化キーブロック（ＥＫＢ）の管理エンティテイに基づいて設定された構成であり、有効化キーブロック（ＥＫＢ）の管理エンティテイ別に独立した検証値を設定し、格納する構成を有することを特徴とする。

【００２０】

さらに、本発明のデータ処理装置の一実施態様において、前記検証値は、検証対象となるコンテンツおよびヘッダ情報等のコンテンツ関連データを構成する部分データをメッセージとしてＤＥＳ暗号処理によって生成されるメッセージ認証符号（ＭＡＣ）に基づいて生成されるデータであることを特徴とする。

【００２１】

さらに、本発明の第２の側面は、

記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行するデータ処理方法であり、

前記検証値を、コンテンツのカテゴリ毎に独立した検証値として生成して格納

することを特徴とするデータ処理方法にある。

【００２２】

さらに、本発明のデータ処理方法の一実施態様において、コンテンツ利用に際して、利用対象コンテンツ構成データに基づいて検証値を算出し、該算出検証値と、予め格納された検証値との比較処理を実行し、一致する場合にのみコンテンツ利用を行なうことを特徴とする。

【００２３】

さらに、本発明のデータ処理方法の一実施態様において、前記記憶装置は、複数のディレクトリの各々に異なるカテゴリのコンテンツデータを格納する構成を有し、前記検証値は、前記複数のディレクトリの各々を単位とするコンテンツの集合に対して生成することを特徴とする。

【００２４】

さらに、本発明のデータ処理方法の一実施態様において、前記記憶装置は、フラッシュメモリであり、前記カテゴリ毎の検証値は、フラッシュメモリの使用禁止ブロックとして設定された領域に格納することを特徴とする。

【００２５】

さらに、本発明のデータ処理方法の一実施態様において、前記カテゴリは、コンテンツの種類に基づいて設定され、コンテンツの種類別に独立した検証値を設定し、格納することを特徴とする。

【００２６】

さらに、本発明のデータ処理方法の一実施態様において、前記カテゴリは、コンテンツの暗号処理鍵として設定されるコンテンツキーＫｃｏｎを暗号化して提供する有効化キーブロック（ＥＫＢ）の管理エンティテイに基づいて設定され、有効化キーブロック（ＥＫＢ）の管理エンティテイ別に独立した検証値を設定し、格納することを特徴とする。

【００２７】

さらに、本発明のデータ処理方法の一実施態様において、前記検証値は、検証対象となるコンテンツおよびヘッダ情報等のコンテンツ関連データを構成する部分データをメッセージとしてＤＥＳ暗号処理によって生成されるメッセージ認証

符号（ＭＡＣ）に基づいて生成されるデータであることを特徴とする。

【００２８】

さらに、本発明の第３の側面は、

記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行するデータ処理をコンピュータ・システム上で実行せしめるコンピュータ・プログラムを提供するプログラム提供媒体であって、前記コンピュータ・プログラムは、

検証値を、コンテンツのカテゴリ毎に独立した検証値として生成して格納するステップを有することを特徴とするプログラム提供媒体にある。

【００２９】

なお、本発明の第３の側面に係るプログラム提供媒体は、例えば、様々なプログラム・コードを実行可能な汎用コンピュータ・システムに対して、コンピュータ・プログラムをコンピュータ可読な形式で提供する媒体である。媒体は、ＣＤやＦＤ、ＭＯなどの記録媒体、あるいは、ネットワークなどの伝送媒体など、その形態は特に限定されない。

【００３０】

このようなプログラム提供媒体は、コンピュータ・システム上で所定のコンピュータ・プログラムの機能を実現するための、コンピュータ・プログラムと提供媒体との構造上又は機能上の協働的関係を定義したものである。換言すれば、該提供媒体を介してコンピュータ・プログラムをコンピュータ・システムにインストールすることによって、コンピュータ・システム上では協働的作用が発揮され、本発明の他の側面と同様の作用効果を得ることができるのである。

【００３１】

本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

【００３２】

【発明の実施の形態】

［システム概要］

図１に本発明のデータ処理システムの適用可能なコンテンツ配信システム例を

示す。コンテンツ配信手段１０は、データ処理手段２０に対して、コンテンツあるいはコンテンツキー、その他、認証処理キー等のデータを暗号化して送信する。データ処理手段２０では、受信した暗号化コンテンツ、あるいは暗号化コンテンツキー等を復号してコンテンツあるいはコンテンツキー等を取得して、画像データ、音声データの再生、あるいは各種プログラムを実行する。コンテンツの配信手段１０とデータ処理手段２０との間のデータ交換は、インターネット等のネットワークを介して、あるいはＤＶＤ、ＣＤ、その他の流通可能な記憶媒体を介して実行される。

【００３３】

データ処理手段２０は、例えばフラッシュメモリ等の記憶手段を備えたメモリーカード等のデータ記憶手段３０にデータを格納して保存する。データ記憶手段３０には、暗号処理機能を有する記憶手段としての例えばメモリーカード（具体例としてはメモリスティック(Memory Stick:商標)）が含まれる。データ処理手段２０からデータ記憶手段３０に対するデータ格納処理、およびデータ記憶手段３０からデータ処理手段に対するデータ移動の際には、相互認証処理、およびデータの暗号処理が実行され不正なデータコピーの防止が図られる。

【００３４】

なお、データ処理手段２０に含まれる各機器間でのコンテンツデータの移動も可能であり、この際にも機器間の相互認証処理、データの暗号処理が実行される。

【００３５】

コンテンツ配信手段１０としては、インターネット１１、衛星放送１２、電話回線１３、ＤＶＤ、ＣＤ等のメディア１４等があり、一方、データ処理手段２０のデバイスとしては、パーソナルコンピュータ（ＰＣ）２１、ポータブルデバイス（ＰＤ）２２、携帯電話、ＰＤＡ（Personal Digital Assistants）等の携帯機器２３、ＤＶＤ、ＣＤプレーヤ等の記録再生器、ゲーム端末２４、メモリーカード（ｅｘ．メモリースティック（商標））を利用した再生装置２５等がある。これらデータ処理手段２０の各デバイスは、コンテンツ配信手段１０から提供されるコンテンツをネットワーク等の通信手段あるいは、他のデータ処理手段、ま

たは、データ記憶手段３０から取得可能である。

【００３６】

図２に、代表的なコンテンツデータの移動処理例を示す。図２に示すシステムは、パーソナルコンピュータ（ＰＣ）１００、再生装置２００および記憶装置３００間でのデータ（コンテンツ）の移動処理例を示した図である。ＰＣ１００は、プログラムおよびデータ記憶用のハードディスク（ＨＤ）を有し、さらに、外部記憶媒体としてのＣＤ，ＤＶＤ等を装着可能な構成を持つ。

【００３７】

パーソナルコンピュータ（ＰＣ）１００は、インターネット、公衆回線等の各種ネットワークに接続可能であり、例えば、ＥＭＤ(Electronic Music Distribution: 電子音楽配信）などのサービスを提供する図示しないサービスプロバイダのホストコンピュータから、ネットワークをしてオーディオデータ、画像データ、プログラム等の各種データを受信し、受信したデータを必要に応じて復号して、再生装置２００に出力する。また、パーソナルコンピュータ（ＰＣ）１００は、コンテンツデータを受信するに当たって、必要に応じて、サービスプロバイダのホストコンピュータとの間で認証処理および課金処理などを行う。また、パーソナルコンピュータ（ＰＣ）１００は、例えば、ＣＤ、ＤＶＤから入力したデータを再生装置２００に出力する。

【００３８】

記憶装置３００は、再生装置２００に対して着脱自在な装置、例えばメモリスティック(Memory Stick:商標）であり、フラッシュメモリなどの書き換え可能な半導体メモリを内蔵している。

【００３９】

図２に示すように、ＰＣ１００、再生装置２００、記憶装置３００間におけるデータ移動、例えば音楽データ、画像データ等のデータ再生、データ記録、データコピー等の処理の際にはデータ移動機器間において、相互認証処理が実行され、不正な機器を用いたデータ移動は防止される。これらの処理については後述する。また、コンテンツデータのネットワークまたは各種記憶媒体を介する配信、また、ＰＣと再生装置相互間、あるいは再生装置とメモリカード等の記憶装置間

でのコンテンツ移動の際にはコンテンツを暗号化することでデータのセキュリティが保全される。

【００４０】

［キー配信構成としてのツリー（木）構造について］

上述のようなコンテンツに対する暗号処理に適用する暗号鍵、例えばコンテンツの暗号処理に適用するコンテンツキー、またはコンテンツキーを暗号化するためのコンテンツキー暗号化キー等の様々な暗号処理キーを、安全に正当なライセンスを持つデバイスに配信する構成として、階層キー・ツリー構成について図３以下を用いて説明する。

【００４１】

図３の最下段に示すナンバ０～１５がコンテンツデータの再生、実行を行なうデータ処理手段２０を構成する個々のデバイス、例えばコンテンツ（音楽データ）再生装置である。すなわち図３に示す階層ツリー（木）構造の各葉(リーフ：leaf)がそれぞれのデバイスに相当する。

【００４２】

各デバイス０～１５は、製造時あるいは出荷時、あるいはその後において、図３に示す階層ツリー（木）構造における、自分のリーフからルートに至るまでのノードに割り当てられた鍵（ノードキー）および各リーフのリーフキーからなるキーセットをメモリに格納する。図３の最下段に示すＫ００００～Ｋ１１１１が各デバイス０～１５にそれぞれ割り当てられたリーフキーであり、最上段のＫＲ（ルートキー）から、最下段から２番目の節（ノード）に記載されたキー：ＫＲ～Ｋ１１１をノードキーとする。

【００４３】

図３に示すツリー構成において、例えばデバイス０はリーフキーＫ００００と、ノードキー：Ｋ０００、Ｋ００、Ｋ０、ＫＲを所有する。デバイス５はＫ０１０１、Ｋ０１０、Ｋ０１、Ｋ０、ＫＲを所有する。デバイス１５は、Ｋ１１１１、Ｋ１１１、Ｋ１１、Ｋ１、ＫＲを所有する。なお、図３のツリーにはデバイスが０～１５の１６個のみ記載され、ツリー構造も４段構成の均衡のとれた左右対称構成として示しているが、さらに多くのデバイスがツリー中に構成され、また

、ツリーの各部において異なる段数構成を持つことが可能である。

【００４４】

また、図３のツリー構造に含まれる各デバイスには、様々な記録媒体、例えば、デバイス埋め込み型あるいはデバイスに着脱自在に構成されたフラッシュメモリ等を使用したメモリカード、ＤＶＤ、ＣＤ、ＭＤ等、様々なタイプの記憶装置を利用可能なデバイスが含まれている。さらに、様々なアプリケーションサービスが共存可能である。このような異なるデバイス、異なるアプリケーションの共存構成の上に図３に示すコンテンツあるいは鍵配布構成である階層ツリー構造が適用される。

【００４５】

これらの様々なデバイス、アプリケーションが共存するシステムにおいて、例えば図３の点線で囲んだ部分、すなわちデバイス０，１，２，３を同一の記録媒体を用いる１つのグループとして設定する。例えば、この点線で囲んだグループ内に含まれるデバイスに対しては、まとめて、共通のコンテンツを暗号化してプロバイダから送付したり、各デバイス共通に使用するコンテンツキーを送付したり、あるいは各デバイスからプロバイダあるいは決済機関等にコンテンツ料金の支払データをやはり暗号化して出力するといった処理が実行される。コンテンツプロバイダ、あるいは決済処理機関等、各デバイスとのデータ送受信を行なう機関は、図３の点線で囲んだ部分、すなわちデバイス０，１，２，３を１つのグループとして一括してデータを送付する処理を実行する。このようなグループは、図３のツリー中に複数存在する。コンテンツプロバイダ、あるいは決済処理機関等、各デバイスとのデータ送受信を行なう機関は、メッセージデータ配信手段として機能する。

【００４６】

なお、ノードキー、リーフキーは、ある１つの鍵管理センタによって統括して管理してもよいし、各グループに対する様々なデータ送受信を行なうプロバイダ、決済機関等のメッセージデータ配信手段によってグループごとに管理する構成としてもよい。これらのノードキー、リーフキーは例えばキーの漏洩等の場合に更新処理が実行され、この更新処理は鍵管理センタ、プロバイダ、決済機関等が

実行する。

【００４７】

このツリー構造において、図３から明らかなように、１つのグループに含まれる３つのデバイス０，１，２，３はノードキーとして共通のキーＫ００、Ｋ０、ＫＲを保有する。このノードキー共有構成を利用することにより、例えば共通のコンテンツキーをデバイス０，１，２，３のみに提供することが可能となる。たとえば、共通に保有するノードキーＫ００自体をコンテンツキーとして設定すれば、新たな鍵送付を実行することなくデバイス０，１，２，３のみに共通のコンテンツキーの設定が可能である。また、新たなコンテンツキーＫｃｏｎをノードキーＫ００で暗号化した値Ｅｎｃ（Ｋ００，Ｋｃｏｎ）を、ネットワークを介してあるいは記録媒体に格納してデバイス０，１，２，３に配布すれば、デバイス０，１，２，３のみが、それぞれのデバイスにおいて保有する共有ノードキーＫ００を用いて暗号Ｅｎｃ（Ｋ００，Ｋｃｏｎ）を解いてコンテンツキー：Ｋｃｏｎを得ることが可能となる。なお、Ｅｎｃ（Ｋａ，Ｋｂ）はＫｂをＫａによって暗号化したデータであることを示す。

【００４８】

また、ある時点ｔにおいて、デバイス３の所有する鍵：Ｋ００１１,Ｋ００１，Ｋ００,Ｋ０,ＫＲが攻撃者（ハッカー）により解析されて露呈したことが発覚した場合、それ以降、システム（デバイス０，１，２，３のグループ）で送受信されるデータを守るために、デバイス３をシステムから切り離す必要がある。そのためには、ノードキー：Ｋ００１,Ｋ００,Ｋ０,ＫＲをそれぞれ新たな鍵Ｋ（ｔ）００１,Ｋ（ｔ）００,Ｋ（ｔ）０,Ｋ（ｔ）Ｒに更新し、デバイス０，１，２にその更新キーを伝える必要がある。ここで、Ｋ（ｔ）ａａａは、鍵Ｋａａａの世代（Generation）：ｔの更新キーであることを示す。

【００４９】

更新キーの配布処理ついて説明する。キーの更新は、例えば、図４（Ａ）に示す有効化キーブロック（ＥＫＢ：Enabling Key Block）と呼ばれるブロックデータによって構成されるテーブルをたとえばネットワーク、あるいは記録媒体に格納してデバイス０，１，２に供給することによって実行される。なお、有効化キ

ーブロック（ＥＫＢ）は、図３に示すようなツリー構造を構成する各リーフに対応するデバイスに新たに更新されたキーを配布するための暗号化キーによって構成される。有効化キーブロック（ＥＫＢ）は、キー更新ブロック（ＫＲＢ：Key Renewal Block）と呼ばれることもある。

【００５０】

図４（Ａ）に示す有効化キーブロック（ＥＫＢ）には、ノードキーの更新の必要なデバイスのみが更新可能なデータ構成を持つブロックデータとして構成される。図４の例は、図３に示すツリー構造中のデバイス０，１，２において、世代ｔの更新ノードキーを配布することを目的として形成されたブロックデータである。図３から明らかなように、デバイス０，デバイス１は、更新ノードキーとしてＫ（ｔ）００、Ｋ（ｔ）０、Ｋ（ｔ）Ｒが必要であり、デバイス２は、更新ノードキーとしてＫ（ｔ）００１、Ｋ（ｔ）００、Ｋ（ｔ）０、Ｋ（ｔ）Ｒが必要である。

【００５１】

図４（Ａ）のＥＫＢに示されるようにＥＫＢには複数の暗号化キーが含まれる。最下段の暗号化キーは、Ｅｎｃ（Ｋ００１０，Ｋ（ｔ）００１）である。これはデバイス２の持つリーフキーＫ００１０によって暗号化された更新ノードキーＫ（ｔ）００１であり、デバイス２は、自身の持つリーフキーによってこの暗号化キーを復号し、Ｋ（ｔ）００１を得ることができる。また、復号により得たＫ（ｔ）００１を用いて、図４（Ａ）の下から２段目の暗号化キーＥｎｃ（Ｋ（ｔ）００１，Ｋ（ｔ）００）を復号可能となり、更新ノードキーＫ（ｔ）００を得ることができる。以下順次、図４（Ａ）の上から２段目の暗号化キーＥｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）０）を復号し、更新ノードキーＫ（ｔ）０、図４（Ａ）の上から１段目の暗号化キーＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）を復号しＫ（ｔ）Ｒを得る。一方、デバイスＫ００００．Ｋ０００１は、ノードキーＫ０００は更新する対象に含まれておらず、更新ノードキーとして必要なのは、Ｋ（ｔ）００、Ｋ（ｔ）０、Ｋ（ｔ）Ｒである。デバイスＫ００００．Ｋ０００１は、図４（Ａ）の上から３段目の暗号化キーＥｎｃ（Ｋ０００，Ｋ（ｔ）００）を復号しＫ（ｔ）００、を取得し、以下、図４（Ａ）の上から２段目の暗号化キーＥｎｃ（

Ｋ（ｔ）００，Ｋ（ｔ）０）を復号し、更新ノードキーＫ（ｔ）０、図４（Ａ）の上から１段目の暗号化キーＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）を復号しＫ（ｔ）Ｒを得る。このようにして、デバイス０，１，２は更新した鍵Ｋ（ｔ）００１,Ｋ（ｔ）００,Ｋ（ｔ）０,Ｋ（ｔ）Ｒを得ることができる。なお、図４（Ａ）のインデックスは、復号キーとして使用するノードキー、リーフキーの絶対番地を示す。

【００５２】

図３に示すツリー構造の上位段のノードキー：Ｋ（ｔ）０,Ｋ（ｔ）Ｒの更新が不要であり、ノードキーＫ００のみの更新処理が必要である場合には、図４（Ｂ）の有効化キーブロック（ＥＫＢ）を用いることで、更新ノードキーＫ（ｔ）００をデバイス０，１，２に配布することができる。

【００５３】

図４（Ｂ）に示すＥＫＢは、例えば特定のグループにおいて共有する新たなコンテンツキーを配布する場合に利用可能である。具体例として、図３に点線で示すグループ内のデバイス０，１，２，３がある記録媒体を用いており、新たな共通のコンテンツキーＫ（ｔ）ｃｏｎが必要であるとする。このとき、デバイス０，１，２，３の共通のノードキーＫ００を更新したＫ（ｔ）００を用いて新たな共通の更新コンテンツキー：Ｋ（ｔ）ｃｏｎを暗号化したデータＥｎｃ（Ｋ（ｔ），Ｋ（ｔ）ｃｏｎ）を図４（Ｂ）に示すＥＫＢとともに配布する。この配布により、デバイス４など、その他のグループの機器においては復号されないデータとしての配布が可能となる。

【００５４】

すなわち、デバイス０，１，２はＥＫＢを処理して得たＫ（ｔ）００を用いて上記暗号文を復号すれば、ｔ時点でのコンテンツキーＫ（ｔ）ｃｏｎを得ることが可能になる。

【００５５】

［ＥＫＢを使用したコンテンツキーの配布］

図５に、ｔ時点でのコンテンツキーＫ（ｔ）ｃｏｎを得る処理例として、Ｋ（ｔ）００を用いて新たな共通のコンテンツキーＫ（ｔ）ｃｏｎを暗号化したデー

タＥｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）ｃｏｎ）と図４（Ｂ）に示すＥＫＢとを記録媒体を介して受領したデバイス０の処理を示す。すなわちＥＫＢによる暗号化メッセージデータをコンテンツキーＫ（ｔ）ｃｏｎとした例である。

【００５６】

図５に示すように、デバイス０は、記録媒体に格納されている世代：t時点のＥＫＢと自分があらかじめ格納しているノードキーＫ０００を用いて上述したと同様のＥＫＢ処理により、ノードキーＫ（ｔ）００を生成する。さらに、復号した更新ノードキーＫ（ｔ）００を用いて更新コンテンツキーＫ（ｔ）ｃｏｎを復号して、後にそれを使用するために自分だけが持つリーフキーＫ００００で暗号化して格納する。

【００５７】

［ＥＫＢのフォーマット］

図６に有効化キーブロック（ＥＫＢ）のフォーマット例を示す。バージョン６０１は、有効化キーブロック（ＥＫＢ）のバージョンを示す識別子である。なお、バージョンは最新のＥＫＢを識別する機能とコンテンツとの対応関係を示す機能を持つ。デプスは、有効化キーブロック（ＥＫＢ）の配布先のデバイスに対する階層ツリーの階層数を示す。データポインタ６０３は、有効化キーブロック（ＥＫＢ）中のデータ部の位置を示すポインタであり、タグポインタ６０４はタグ部の位置、署名ポインタ６０５は署名の位置を示すポインタである。

【００５８】

データ部６０６は、例えば更新するノードキーを暗号化したデータを格納する。例えば図５に示すような更新されたノードキーに関する各暗号化キー等を格納する。

【００５９】

タグ部６０７は、データ部に格納された暗号化されたノードキー、リーフキーの位置関係を示すタグである。このタグの付与ルールを図７を用いて説明する。図７では、データとして先に図４（Ａ）で説明した有効化キーブロック（ＥＫＢ）を送付する例を示している。この時のデータは、図７の表（ｂ）に示すようになる。このときの暗号化キーに含まれるトップノードのアドレスをトップノード

アドレスとする。この場合は、ルートキーの更新キーＫ（ｔ）Ｒが含まれているので、トップノードアドレスはＫＲとなる。このとき、例えば最上段のデータＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）は、図７の（ａ）に示す階層ツリーに示す位置にある。ここで、次のデータは、Ｅｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）０）であり、ツリー上では前のデータの左下の位置にある。データがある場合は、タグが０、ない場合は１が設定される。タグは｛左（Ｌ）タグ，右（Ｒ）タグ｝として設定される。最上段のデータＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）の左にはデータがあるので、Ｌタグ＝０、右にはデータがないので、Ｒタグ＝１となる。以下、すべてのデータにタグが設定され、図７（ｃ）に示すデータ列、およびタグ列が構成される。

【００６０】

タグは、データＥｎｃ（Ｋｘｘｘ，Ｋｙｙｙ）がツリー構造のどこに位置しているのかを示すために設定されるものである。データ部に格納されるキーデータＥｎｃ（Ｋｘｘｘ，Ｋｙｙｙ）．．．は、単純に暗号化されたキーの羅列データに過ぎないので、上述したタグによってデータとして格納された暗号化キーのツリー上の位置を判別可能としたものである。上述したタグを用いずに、先の図４で説明した構成のように暗号化データに対応させたノード・インデックスを用いて、例えば、

０：Ｅｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）ｒｏｏｔ）

００：Ｅｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）０）

０００：Ｅｎｃ（Ｋ（（ｔ）０００，Ｋ（Ｔ）００）

．．．

のようなデータ構成とすることも可能であるが、このようなインデックスを用いた構成とすると冗長なデータとなりデータ量が増大し、ネットワークを介する配信等においては好ましくない。これに対し、上述したタグをキー位置を示す索引データとして用いることにより、少ないデータ量でキー位置の判別が可能となる。

【００６１】

図６に戻って、ＥＫＢフォーマットについてさらに説明する。署名（Signatur

e）は、有効化キーブロック（ＥＫＢ）を発行した例えば鍵管理センタ、コンテンツロバイダ、決済機関等が実行する電子署名である。ＥＫＢを受領したデバイスは署名検証によって正当な有効化キーブロック（ＥＫＢ）発行者が発行した有効化キーブロック（ＥＫＢ）であることを確認する。

【００６２】

［ＥＫＢを使用したコンテンツキーおよびコンテンツの配信］

上述の例では、コンテンツキーのみをＥＫＢとともに送付する例について説明したが、コンテンツキーで暗号化したコンテンツと、コンテンツキー暗号キーで暗号化したコンテンツキーと、ＥＫＢによって暗号化したコンテンツキー暗号キーを併せて送付する構成について以下説明する。

【００６３】

図８にこのデータ構成を示す。図８（ａ）に示す構成において、Ｅｎｃ（Ｋｃｏｎ，ｃｏｎｔｅｎｔ）８０１は、コンテンツ（Content）をコンテンツキー（Ｋｃｏｎ）で暗号化したデータであり、Ｅｎｃ（ＫＥＫ，Ｋｃｏｎ）８０２は、コンテンツキー（Ｋｃｏｎ）をコンテンツキー暗号キー（ＫＥＫ：Key Encryption Key）で暗号化したデータであり、Ｅｎｃ（ＥＫＢ，ＫＥＫ）８０３は、コンテンツキー暗号キーＫＥＫを有効化キーブロック（ＥＫＢ）によって暗号化したデータであることを示す。

【００６４】

ここで、コンテンツキー暗号キーＫＥＫは、図３で示すノードキー（Ｋ０００，Ｋ００…）、あるいはルートキー（ＫＲ）自体であってもよく、またノードキー（Ｋ０００，Ｋ００…）、あるいはルートキー（ＫＲ）によって暗号化されたキーであってもよい。

【００６５】

図８（ｂ）は、複数のコンテンツがメディアに記録され、それぞれが同じＥｎｃ（ＥＫＢ，ＫＥＫ）８０５を利用している場合の構成例を示す、このような構成においては、各データに同じＥｎｃ（ＥＫＢ，ＫＥＫ）を付加することなく、Ｅｎｃ（ＥＫＢ，ＫＥＫ）にリンクするリンク先を示すデータを各データに付加する構成とすることができる。

【００６６】

図９にコンテンツキー暗号キーＫＥＫを、図３に示すノードキーＫ００を更新した更新ノードキーＫ（ｔ）００として構成した場合の例を示す。この場合、図３の点線枠で囲んだグループにおいてデバイス３が、例えば鍵の漏洩によりリボーク（排除）されているとして、他のグループのメンバ、すなわち、デバイス０，１，２に対して図９に示す（ａ）有効化キーブロック（ＥＫＢ）と、（ｂ）コンテンツキー（Ｋｃｏｎ）をコンテンツキー暗号キー（ＫＥＫ＝Ｋ（ｔ）００）で暗号化したデータと、（ｃ）コンテンツ（content）をコンテンツキー（Ｋｃｏｎ）で暗号化したデータとを配信することにより、デバイス０，１，２はコンテンツを得ることができる。

【００６７】

図９の右側には、デバイス０における復号手順を示してある。デバイス０は、まず、受領した有効化キーブロックから自身の保有するリーフキーＫ０００を用いた復号処理により、コンテンツキー暗号キー（ＫＥＫ＝Ｋ（ｔ）００）を取得する。次に、Ｋ（ｔ）００による復号によりコンテンツキーＫｃｏｎを取得し、さらにコンテンツキーＫｃｏｎによりコンテンツの復号を行なう。これらの処理により、デバイス０はコンテンツを利用可能となる。デバイス１，２においても各々異なる処理手順でＥＫＢを処理することにより、コンテンツキー暗号キー（ＫＥＫ＝Ｋ（ｔ）００）を取得することが可能となり、同様にコンテンツを利用することが可能となる。

【００６８】

図３に示す他のグループのデバイス４，５，６…は、この同様のデータ（ＥＫＢ）を受信したとしても、自身の保有するリーフキー、ノードキーを用いてコンテンツキー暗号キー（ＫＥＫ＝Ｋ（ｔ）００）を取得することができない。同様にリボークされたデバイス３においても、自身の保有するリーフキー、ノードキーでは、コンテンツキー暗号キー（ＫＥＫ＝Ｋ（ｔ）００）を取得することができず、正当な権利を有するデバイスのみがコンテンツを復号して利用することが可能となる。

【００６９】

このように、ＥＫＢを利用したコンテンツキーの配送を用いれば、データ量を少なくして、かつ安全に正当権利者のみが復号可能とした暗号化コンテンツを配信することが可能となる。

【００７０】

なお、有効化キーブロック（ＥＫＢ）、コンテンツキー、暗号化コンテンツ等は、ネットワークを介して安全に配信することが可能な構成であるが、有効化キーブロック（ＥＫＢ）、コンテンツキー、暗号化コンテンツをＤＶＤ、ＣＤ等の記録媒体に格納してユーザに提供することも可能である。この場合、記録媒体に格納された暗号化コンテンツの復号には、同一の記録媒体に格納された有効化キーブロック（ＥＫＢ）の復号により得られるコンテンツキーを使用するように構成すれば、予め正当権利者のみが保有するリーフキー、ノードキーによってのみ利用可能な暗号化コンテンツの配布処理、すなわち利用可能なユーザデバイスを限定したコンテンツ配布が簡易な構成で実現可能となる。

【００７１】

図１０に記録媒体に暗号化コンテンツとともに有効化キーブロック（ＥＫＢ）を格納した構成例を示す。図１０に示す例においては、記録媒体にコンテンツＣ１～Ｃ４が格納され、さらに各格納コンテンツに対応するの有効化キーブロック（ＥＫＢ）を対応付けたデータが格納され、さらにバージョンＭの有効化キーブロック（ＥＫＢ＿Ｍ）が格納されている。例えばＥＫＢ＿１はコンテンツＣ１を暗号化したコンテンツキーＫｃｏｎ１を生成するのに使用され、例えばＥＫＢ＿２はコンテンツＣ２を暗号化したコンテンツキーＫｃｏｎ２を生成するのに使用される。この例では、バージョンＭの有効化キーブロック（ＥＫＢ＿Ｍ）が記録媒体に格納されており、コンテンツＣ３，Ｃ４は有効化キーブロック（ＥＫＢ＿Ｍ）に対応付けられているので、有効化キーブロック（ＥＫＢ＿Ｍ）の復号によりコンテンツＣ３，Ｃ４のコンテンツキーを取得することができる。ＥＫＢ＿１、ＥＫＢ＿２はディスクに格納されていないので、新たな提供手段、例えばネットワーク配信、あるいは記録媒体による配信によってそれぞれのコンテンツキーを復号するために必要なＥＫＢ＿１，ＥＫＢ＿２を取得することが必要となる。

【００７２】

［階層ツリー構造のカテゴリー分類］

暗号鍵をルートキー、ノードキー、リーフキー等、図３の階層ツリー構造として構成し、コンテンツキー、認証キー、ＩＣＶ生成キー、あるいはプログラムコード、データ等を有効化キーブロック（ＥＫＢ）とともに暗号化して配信する構成について説明してきたが、ノードキー等を定義している階層ツリー構造を各デバイスのカテゴリー毎に分類して効率的なキー更新処理、暗号化キー配信、データ配信を実行する構成について、以下説明する。

【００７３】

図１１に階層ツリー構造のカテゴリーの分類の一例を示す。図１１において、階層ツリー構造の最上段には、ルートキーＫｒｏｏｔ１１０１が設定され、以下の中間段にはノードキー１１０２が設定され、最下段には、リーフキー１１０３が設定される。各デバイスは個々のリーフキーと、リーフキーからルートキーに至る一連のノードキー、ルートキーを保有する。

【００７４】

ここで、一例として最上段から第M段目のあるノードをカテゴリノード１１０４として設定する。すなわち第M段目のノードの各々を特定カテゴリのデバイス設定ノードとする。第M段の１つのノードを頂点として以下、Ｍ＋１段以下のノード、リーフは、そのカテゴリに含まれるデバイスに関するノードおよびリーフとする。

【００７５】

例えば図１１の第M段目の１つのノード１１０５にはカテゴリ［メモリステッイク（商標）］が設定され、このノード以下に連なるノード、リーフはメモリステッイクを使用した様々なデバイスを含むカテゴリ専用のノードまたはリーフとして設定される。すなわち、ノード１１０５以下を、メモリスティックのカテゴリに定義されるデバイスの関連ノード、およびリーフの集合として定義する。

【００７６】

さらに、M段から数段分下位の段をサブカテゴリノード１１０６として設定することができる。例えば図に示すようにカテゴリ［メモリスティック］ノード１１０５の２段下のノードに、メモリスティックを使用したデバイスのカテゴリに

含まれるサブカテゴリノードとして、［再生専用器］のノードを設定する。さらに、サブカテゴリノードである再生専用器のノード１１０６以下に、再生専用器のカテゴリに含まれる音楽再生機能付き電話のノード１１０７が設定され、さらにその下位に、音楽再生機能付き電話のカテゴリに含まれる［ＰＨＳ］ノード１１０８と［携帯電話］ノード１１０９を設定することができる。

【００７７】

さらに、カテゴリ、サブカテゴリは、デバイスの種類のみならず、例えばあるメーカー、コンテンツプロバイダ、決済機関等が独自に管理するノード、すなわち処理単位、管轄単位、あるいは提供サービス単位等、任意の単位（これらを総称して以下、エンティティと呼ぶ）で設定することが可能である。例えば１つのカテゴリノードをゲーム機器メーカーの販売するゲーム機器ＸＹＺ専用の頂点ノードとして設定すれば、メーカーの販売するゲーム機器ＸＹＺにその頂点ノード以下の下段のノードキー、リーフキーを格納して販売することが可能となり、その後、暗号化コンテンツの配信、あるいは各種キーの配信、更新処理を、その頂点ノードキー以下のノードキー、リーフキーによって構成される有効化キーブロック（ＥＫＢ）を生成して配信し、頂点ノード以下のデバイスに対してのみ利用可能なデータが配信可能となる。

【００７８】

このように、１つのノードを頂点としして、以下のノードをその頂点ノードに定義されたカテゴリ、あるいはサブカテゴリの関連ノードとして設定する構成とすることにより、カテゴリ段、あるいはサブカテゴリ段の１つの頂点ノードを管理するメーカー、コンテンツプロバイダ等がそのノードを頂点とする有効化キーブロック（ＥＫＢ）を独自に生成して、頂点ノード以下に属するデバイスに配信する構成が可能となり、頂点ノードに属さない他のカテゴリのノードに属するデバイスには全く影響を及ぼさずにキー更新を実行することができる。

【００７９】

［簡略ＥＫＢによるキー配信構成］

先に説明した例えば図３のツリー構成において、キー、例えばコンテンツキーを所定デバイス（リーフ）宛に送付する場合、キー配布先デバイスの所有してい

るリーフキー、ノードキーを用いて復号可能な有効化キーブロック（ＥＫＢ）を生成して提供する。例えば図１２（ａ）に示すツリー構成において、リーフを構成するデバイスａ，ｇ，ｊに対してキー、例えばコンテンツキーを送信する場合、ａ，ｇ，ｊの各ノードにおいて復号可能な有効化キーブロック（ＥＫＢ）を生成して配信する。

【００８０】

例えば更新ルートキーＫ（ｔ）ｒｏｏｔでコンテンツキーＫ（ｔ）ｃｏｎを暗号化処理し、ＥＫＢとともに配信する場合を考える。この場合、デバイスａ，ｇ，ｊは、それぞれが図１２（ｂ）に示すリーフおよびノードキーを用いて、ＥＫＢの処理を実行してＫ（ｔ）ｒｏｏｔを取得し、取得した更新ルートキーＫ（ｔ）ｒｏｏｔによってコンテンツキーＫ（ｔ）ｃｏｎの復号処理を実行してコンテンツキーを得る。

【００８１】

この場合に提供される有効化キーブロック（ＥＫＢ）の構成は、図１３に示すようになる。図１３に示す有効化キーブロック（ＥＫＢ）は、先の図６で説明した有効化キーブロック（ＥＫＢ）のフォーマットにしたがって構成されたものであり、データ（暗号化キー）と対応するタグとを持つ。タグは、先に図７を用いて説明したように左（Ｌ）、右（Ｒ）、それぞれの方向にデータがあれば０、無ければ１を示している。

【００８２】

有効化キーブロック（ＥＫＢ）を受領したデバイスは、有効化キーブロック（ＥＫＢ）の暗号化キーとタグに基づいて、順次暗号化キーの復号処理を実行して上位ノードの更新キーを取得していく。図１３に示すように、有効化キーブロック（ＥＫＢ）は、ルートからリーフまでの段数（デプス）が多いほど、そのデータ量は増加していく。段数（デプス）は、デバイス（リーフ）数に応じて増大するものであり、キーの配信先となるデバイス数が多い場合は、ＥＫＢのデータ量がさらに増大することになる。

【００８３】

このような有効化キーブロック（ＥＫＢ）のデータ量の削減を可能とした構成

について説明する。図１４は、有効化キーブロック（ＥＫＢ）をキー配信デバイスに応じて簡略化して構成した例を示すものである。

【００８４】

図１３と同様、リーフを構成するデバイスａ，ｇ，ｊに対してキー、例えばコンテンツキーを送信する場合を想定する。図１４の（ａ）に示すように、キー配信デバイスによってのみ構成されるツリーを構築する。この場合、図１２（ｂ）に示す構成に基づいて新たなツリー構成として図１４（ｂ）のツリー構成が構築される。ＫｒｏｏｔからＫｊまでは全く分岐がなく１つの枝のみが存在すればよく、ＫｒｏｏｔからＫａおよびＫｇに至るためには、Ｋ０に分岐点を構成するのみで、２分岐構成の図１４（ａ）のツリーが構築される。

【００８５】

図１４（ａ）に示すように、ノードとしてＫ０のみを持つ簡略化したツリーが生成される。更新キー配信のための有効化キーブロック（ＥＫＢ）は、これらの簡略ツリーに基づいて生成する。図１４（ａ）に示すツリーは、有効化キーブロック（ＥＫＢ）を復号可能な末端ノードまたはリーフを最下段とした２分岐型ツリーを構成するパスを選択して不要ノードを省略することにより再構築される再構築階層ツリーである。更新キー配信のための有効化キーブロック（ＥＫＢ）は、この再構築階層ツリーのノードまたはリーフに対応するキーのみに基づいて構成される。

【００８６】

先の図１３で説明した有効化キーブロック（ＥＫＢ）は、各リーフａ，ｇ，ｊからＫｒｏｏｔに至るまでのすべてのキーを暗号化したデータを格納していたが、簡略化ＥＫＢは、簡略化したツリーを構成するノードについてのみの暗号化データを格納する。図１４（ｂ）に示すようにタグは３ビット構成を有する。第１および第２ビットは、図１３の例と、同様の意味を持ち、左（Ｌ）、右（Ｒ）、それぞれの方向にデータがあれば０、無ければ１を示す。第３番目のビットは、ＥＫＢ内に暗号化キーが格納されているか否かを示すためのビットであり、データが格納されている場合は１、データが無い場合は、０として設定される。

【００８７】

データ通信網、あるいは記憶媒体に格納されてデバイス（リーフ）に提供される有効化キーブロック（ＥＫＢ）は、図１４（ｂ）に示すように、図１３に示す構成に比較すると、データ量が大幅に削減されたものとなる。図１４に示す有効化キーブロック（ＥＫＢ）を受領した各デバイスは、タグの第３ビットに１が格納された部分のデータのみを順次復号することにより、所定の暗号化キーの復号を実現することができる。例えばデバイスａは、暗号化データＥｎｃ（Ｋａ，Ｋ（ｔ）０）をリーフキーＫａで復号して、ノードキーＫ（ｔ）０を取得して、ノードキーＫ（ｔ）０によって暗号化データＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）ｒｏｏｔ）を復号してＫ（ｔ）ｒｏｏｔを取得する。デバイスｊは、暗号化データＥｎｃ（Ｋｊ，Ｋ（ｔ）ｒｏｏｔ）をリーフキーＫｊで復号して、Ｋ（ｔ）ｒｏｏｔを取得する。

【００８８】

このように、配信先のデバイスによってのみ構成される簡略化した新たなツリー構成を構築して、構築されたツリーを構成するリーフおよびノードのキーのみを用いて有効化キーブロック（ＥＫＢ）を生成することにより、少ないデータ量の有効化キーブロック（ＥＫＢ）を生成することが可能となり、有効化キーブロック（ＥＫＢ）のデータ配信が効率的に実行可能となる。

【００８９】

なお、簡略化した階層ツリー構成は、後段で説明するエンティテイ単位のＥＫＢ管理構成において特に有効に活用可能である。エンティテイは、キー配信構成としてのツリー構成を構成するノードあるいはリーフから選択した複数のノードあるいはリーフの集合体ブロックである。エンティテイは、デバイスの種類に応じて設定される集合であったり、あるいはデバイス提供メーカー、コンテンツプロバイダ、決済機関等の管理単位等、ある共通点を持った処理単位、管轄単位、あるいは提供サービス単位等、様々な態様の集合として設定される。１つのエンティテイには、ある共通のカテゴリに分類されるデバイスが集まっており、例えば複数のエンティテイの頂点ノード（サブルート）によって上述したと同様の簡略化したツリーを再構築してＥＫＢを生成することにより、選択されたエンティテイに属するデバイスにおいて復号可能な簡略化された有効化キーブロック（Ｅ

ＫＢ）の生成、配信が可能となる。エンティテイ単位の管理構成については後段で詳細に説明する。

【００９０】

なお、このような有効化キーブロック（ＥＫＢ）は、光ディスク、ＤＶＤ等の情報記録媒体に格納した構成とすることが可能である。例えば、上述の暗号化キーデータによって構成されるデータ部と、暗号化キーデータの階層ツリー構造における位置識別データとしてのタグ部とを含む有効化キーブロック（ＥＫＢ）にさらに、更新ノードキーによって暗号化したコンテンツ等のメッセージデータとを格納した情報記録媒体を各デバイスに提供する構成が可能である。デバイスは有効化キーブロック（ＥＫＢ）に含まれる暗号化キーデータをタグ部の識別データにしたがって順次抽出して復号し、コンテンツの復号に必要なキーを取得してコンテンツの利用を行なうことが可能となる。もちろん、有効化キーブロック（ＥＫＢ）をインターネット等のネットワークを介して配信する構成としてもよい。

【００９１】

[暗号処理機能を有する記憶装置とデータ処理装置間のデータ移動]

次に、上述した階層ツリー構成を適用した有効化キーブロック（ＥＫＢ）によって配信される暗号処理キーを適用した処理構成について、暗号処理機能を有する記憶装置、例えばメモリスティック（商標）等のメモリカードと、データ再生装置間におけるデータ移動処理を中心として説明する。

【００９２】

図１５は、相互にコンテンツデータの移動を実行可能な再生装置と暗号処理機能を有するメモリカード等の記憶装置の詳細構成を示したブロック図である。

【００９３】

図１５に示すように、記憶装置３００は、例えば、主制御モジュール３１、通信インターフェイス３２、制御モジュール３３、フラッシュメモリ３４およびフラッシュメモリ管理モジュール３５を有する。以下、各モジュールについて説明する。

【００９４】

〔制御モジュール３３〕

図１５に示すように、制御モジュール３３は、例えば、乱数発生ユニット５０、記憶ユニット５１、鍵生成／演算ユニット５２、相互認証ユニット５３、暗号化／復号ユニット５４および制御ユニット５５を有する。制御モジュール３３は、シングルチップの暗号処理専用の集積回路であり、多層構造を有し、内部のメモリセルはアルミニウム層などのダミー層に挟まれている。また、制御モジュール３３は、動作電圧または動作周波数の幅が狭く、外部から不正にデータを読み出せないように耐タンパー性を有している。乱数発生ユニット５０は、乱数発生指示を受けると、６４ビット（８バイト）の乱数を発生する。

【００９５】

記憶ユニット５１は、例えば、ＥＥＰＲＯＭ(Electrically Erasable Programmable Read Only Memory) などの不揮発性メモリであり、認証処理に必要な鍵データなどの種々のデータを記憶している。図１６は、記憶ユニット５１に記憶されているデータを説明するための図である。図１６に示すように、記憶ユニット５１は、認証鍵データＩＫ0～ＩＫ31、装置識別データＩＤmおよび記憶用鍵データＫｓｔｒを記憶している。

【００９６】

認証鍵データＩＫ0～ＩＫ31は、記憶装置３００が再生装置２００との間で相互認証を行う際に用いられる鍵データであり、後述するように相互認証を行う度に認証鍵データＩＫ0～ＩＫ31のうち一の認証鍵データがランダムに選択される。なお、認証鍵データＩＫ0～ＩＫ31および記憶用鍵データＫｓｔｒは、記憶装置３００の外部から読めないようになっている。装置識別データＩＤmは、記憶装置３００に対してユニークに付けられた識別データであり、後述するように、記憶装置３００が再生装置２００との間で相互認証を行う際に読み出されて再生装置２００に出力される。記憶用鍵データＫｓｔｒは、後述するように、コンテンツの暗号化に用いられるコンテンツ鍵データＣＫを暗号化してフラッシュメモリ３４に記憶する際に用いられる。

【００９７】

鍵生成／演算ユニット５２は、例えば、ＩＳＯ／ＩＥＣ９７９７のＭＡＣ(Mes

sage Authentication Code) 演算などの種々の演算を行って鍵データを生成する。このとき、ＭＡＣ演算には、例えば、"Block cipher Algorithm"としてＦＩＰＳＰＵＢ４６－２に規定されるＤＥＳ(Data Encryption Standard)が用いられる。ＭＡＣ演算は、任意の長さのデータを固定の長さに圧縮する一方向性ハッシュ関数演算であり、関数値が秘密鍵に依存して定まる。

【００９８】

相互認証ユニット５３は、再生装置２００からオーディオデータを入力してフラッシュメモリ３４に書き込む動作を行うのに先立って、再生装置２００との間で相互認証処理を行う。また、相互認証ユニット５３は、フラッシュメモリ３４からオーディオデータを読み出して再生装置２００に出力する動作を行うのに先立って、再生装置２００との間で相互認証処理を行う。また、相互認証ユニット５３は、相互認証処理において、前述したＭＡＣ演算を行う。当該相互認証処理では、記憶ユニット５１に記憶されているデータが用いられる。

【００９９】

暗号化／復号ユニット５４は、ＤＥＳ、ＩＤＥＡ、ＭＩＳＴＹなどのブロック暗号アルゴリズムでの暗号化を行う。使用するモードは、ＦＩＰＳ　ＰＵＢ８１"ＤＥＳ　ＭＯＤＥＳ　ＯＦ　ＯＰＥＲＡＴＩＯＮ"に規定されているようなＥＣＢ(Electronic Code Book)モードおよびＣＢＣ(Cipher Block Chaining) モードである。また、暗号化／復号ユニット５４は、ＤＥＳ、ＩＤＥＡ、ＭＩＳＴＹなどのブロック復号アルゴリズムでの復号を行う。使用するモードは、上記ＥＣＢモードおよびＣＢＣモードである。当該ＥＣＢモードおよびＣＢＣモードのブロック暗号化／復号では、指定された鍵データを用いて指定されたデータを暗号化／復号する。制御ユニット５５は、乱数発生ユニット５０、記憶ユニット５１、鍵生成／演算ユニット５２、相互認証ユニット５３および暗号化／復号ユニット５４の処理を統括して制御する。

【０１００】

〔フラッシュメモリ３４〕

フラッシュメモリ３４は、例えば、３２Ｍバイトの記憶容量を有する。フラッシュメモリ３４には、相互認証ユニット５３による再生装置２００と記憶装置３

００との間の相互認証処理によって双方が正当な装置であると認められたときに、再生装置２００から入力したオーディオデータあるいは画像データ等、各種データが書き込まれる。また、フラッシュメモリ３４からは、相互認証ユニット５３による再生装置２００と記憶装置３００との間の相互認証処理によって正当な相手であると認められたときに、オーディオデータ、画像データ等が読み出されて再生装置２００に出力される。

【０１０１】

以下、フラッシュメモリ３４に記憶されるデータおよびそのフォーマットについて説明する。図１７は、フラッシュメモリ３４に記憶されるデータを説明するための図である。図１７に示すように、フラッシュメモリ３４には、例えば、再生管理ファイル、複数のトラックデータ（再生データ）ファイルが記憶されている。ここで、再生管理ファイルはトラックデータファイルの再生を管理する管理データを有し、トラックデータファイルはそれぞれ対応するトラックデータ（オーディオデータ）を有している。なお、本実施形態では、トラックデータは、例えば、１曲分のオーディオデータを意味する。以下、フラッシュメモリ３４に記憶されるデータをオーディオデータとした場合の例について説明する。

【０１０２】

図１８は、再生管理ファイルの構成を示し、図１９が一つ（１曲）のＡＴＲＡＣ３データファイルの構成を示す。再生管理ファイルは、１６ＫＢ固定長のファイルである。ＡＴＲＡＣ３データファイルは、曲単位でもって、先頭の属性ヘッダと、それに続く実際の暗号化された音楽データとからなる。属性ヘッダも１６ＫＢ固定長とされ、再生管理ファイルと類似した構成を有する。

【０１０３】

再生管理ファイルは、ヘッダ、１バイトコードのメモリカードの名前ＮＭ１－Ｓ、２バイトコードのメモリカードの名前ＮＭ２－Ｓ、曲順の再生テーブルＴＲＫＴＢＬ、メモリカード全体の付加情報ＩＮＦ－Ｓとからなる。データファイルの先頭の属性ヘッダは、ヘッダ、１バイトコードの曲名ＮＭ１、２バイトコードの曲名ＮＭ２、トラックのキー情報等のトラック情報ＴＲＫＩＮＦ、パーツ情報ＰＲＴＩＮＦと、トラックの付加情報ＩＮＦとからなる。ヘッダには、総パーツ

数、名前の属性、付加情報のサイズの情報等が含まれる。

【０１０４】

属性ヘッダに対してＡＴＲＡＣ３の音楽データが続く。音楽データは、１６ＫＢのブロック毎に区切られ、各ブロックの先頭にヘッダが付加されている。ヘッダには、暗号を復号するための初期値が含まれる。なお、暗号化の処理を受けるのは、ＡＴＲＡＣ３データファイル中の音楽データ等のコンテンツデータのみであって、それ以外の再生管理ファイル、ヘッダ等のデータは、暗号化されない。

【０１０５】

図２０に、ＡＴＲＡＣ３データファイルＡ３Ｄｎｎｎｎのデータ配列例を示す。図２０には、データファイルの属性ヘッダ（１ブロック）と、音楽データファイル（１ブロック）とが示されている。図２０では、この２ブロック（１６×２＝３２Ｋバイト）の各スロットの先頭のバイト（０ｘ０００００～０ｘ７ＦＦ０）が示されている。図２１に分離して示すように、属性ヘッダの先頭から３２バイトがヘッダであり、２５６バイトが曲名領域ＮＭ１（２５６バイト）であり、５１２バイトが曲名領域ＮＭ２（５１２バイト）である。属性ヘッダのヘッダには、下記のデータが書かれる。

【０１０６】

ＢＬＫＩＤ－ＨＤ０（４バイト）

意味：BLOCKID FILE ID

機能：ＡＴＲＡＣ３データファイルの先頭であることを識別するための値

値：固定値＝”ＨＤ＝０”（例えば０ｘ４８４４２Ｄ３０）

【０１０７】

ＭＣｏｄｅ（２バイト）

意味：MAKER CODE

機能：記録した機器の、メーカー、モデルを識別するコード

値：上位１０ビット（メーカーコード）　下位６ビット（機種コード）

【０１０８】

ＢＬＯＣＫ　ＳＥＲＩＡＬ（４バイト）

意味：トラック毎に付けられた連続番号

機能：ブロックの先頭は０から始まり次のブロックは＋１づつインクリメント

編集されても値を変化させない

値：０より始まり０ｘＦＦＦＦＦＦＦＦまで。

【０１０９】

Ｎ１Ｃ＋Ｌ（２バイト）

意味：トラック（曲名）データ（ＮＭ１）の属性を表す

機能：ＮＭ１に使用される文字コードと言語コードを各１バイトで表す

値：ＳＮ１Ｃ＋Ｌと同一

【０１１０】

Ｎ２Ｃ＋Ｌ（２バイト）

意味：トラック（曲名）データ（ＮＭ２）の属性を表す

機能：ＮＭ２に使用される文字コードと言語コードを各１バイトで表す

値：ＳＮ１Ｃ＋Ｌと同一

【０１１１】

ＩＮＦＳＩＺＥ（２バイト）

意味：トラックに関する付加情報の全てを合計したサイズを表す

機能：データサイズを１６バイト単位の大きさで記述、無い場合は必ずオールゼロとすること

値：サイズは０ｘ０００００から０ｘ３Ｃ６（９６６）

【０１１２】

Ｔ－ＰＲＴ（２バイト）

意味：トータルパーツ数

機能：トラックを構成するパーツ数を表す。通常は１

値：１から０ｘ２８５（６４５dec ）

【０１１３】

Ｔ－ＳＵ（４バイト）

意味：トータルＳＵ（サウンドユニット）数、ＳＵは、パーツの最小単位であり、且つＡＴＲＡＣ３でオーディオデータを圧縮する時の最小のデータ単位である。４４．１ｋHzのサンプリング周波数で得られた１０２４サンプル分（１０２

４×１６ビット×２チャンネル）のオーディオデータを約１／１０に圧縮した数百バイトのデータがＳＵである。１ＳＵは、時間に換算して約２３ｍ秒になる。通常は、数千に及ぶＳＵによって１つのパーツが構成される。１クラスタが４２個のＳＵで構成される場合、１クラスタで約１秒の音を表すことができる。１つのトラックを構成するパーツの数は、付加情報サイズに影響される。パーツ数は、１ブロックの中からヘッダや曲名、付加情報データ等を除いた数で決まるために、付加情報が全く無い状態が最大数（６４５個）のパーツを使用できる条件となる。

機能：１トラック中の実際の総ＳＵ数を表す。曲の演奏時間に相当する

値：０ｘ０１から０ｘ００１ＦＦＦＦＦ

【０１１４】

ＩＮＸ（２バイト）（Ｏｐｔｉｏｎ）

意味：INDEX の相対場所

機能：曲のさびの部分（特徴的な部分）の先頭を示すポインタ。曲の先頭からの位置をＳＵの個数を１／４した数で指定する。これは、通常のＳＵの４倍の長さの時間（約９３ｍ秒）に相当する

値：０から０ｘＦＦＦＦ（最大、約６０８４秒）

【０１１５】

ＸＴ（２バイト）（Ｏｐｔｉｏｎ）

意味：INDEX の再生時間

機能：ＩＮＸ-nnnで指定された先頭から再生すべき時間のＳＵの個数を１／４した数で指定する。これは、通常のＳＵの４倍の長さの時間（約９３ｍ秒）に相当する

値：０ｘ０００：無設定　　０ｘ０１から０ｘＦＦＦＥ（最大６０８４秒）

０ｘＦＦＦＦ：曲の終わりまで。

【０１１６】

次に曲名領域ＮＭ１およびＮＭ２について説明する。

【０１１７】

ＮＭ１

意味：曲名を表す文字列

機能：１バイトの文字コードで表した可変長の曲名（最大で２５６）

名前データの終了は、必ず終端コード（０ｘ００）を書き込むこと

サイズはこの終端コードから計算すること、データの無い場合は少なくとも先頭（０ｘ００２０）からヌル（０ｘ００）を１バイト以上記録すること

値：各種文字コード

【０１１８】

ＮＭ２

意味：曲名を表す文字列

機能：２バイトの文字コードで表した可変長の名前データ（最大で５１２）

名前データの終了は、必ず終端コード（０ｘ００）を書き込むこと

サイズはこの終端コードから計算すること、データの無い場合は少なくとも先頭（０ｘ０１２０）からヌル（０ｘ００）を２バイト以上記録すること

値：各種文字コード。

【０１１９】

属性ヘッダの固定位置（０ｘ３２０）から始まる、８０バイトのデータをトラック情報領域ＴＲＫＩＮＦと呼び、主としてセキュリティ関係、コピー制御関係の情報を一括して管理する。図２２にＴＲＫＩＮＦの部分を示す。ＴＲＫＩＮＦ内のデータについて、配置順序に従って以下に説明する。

【０１２０】

ＥＫＩ（１バイト）

意味：前述の階層ツリー構成による有効化キーブロック（ＥＫＢ）によって提供される暗号化コンテンツキー：Ｅ（ＫＥＫｎ，Ｋｃｏｎ）を有するか否かを示す。

機能：ｂｉｔ７＝１でキー有、ｂｉｔ７＝０で無し。ｂｉｔ７＝０の場合は、ＥＫＢ＿ｖｅｒｓｉｏｎ、Ｅ（ＫＥＫｎ，Ｋｃｏｎ）は非参照。

値：０から０ｘＦＦまで

【０１２１】

ＥＫ＿ｖｅｒｓｉｏｎ（４バイト）

意味：前述の階層ツリー構成による有効化キーブロック（ＥＫＢ）によって提供されるコンテンツキーの世代番号、および／または有効化キーブロック（ＥＫＢ）のファイル名を示す。

機能：階層ツリー構成による有効化キーブロック（ＥＫＢ）によって提供されるコンテンツキーを求めるための有効化キーブロック（ＥＫＢ）を示す。

値：０から０ｘＦＦまで

【０１２２】

Ｅ（Ｋｓｔｒ，Ｋｃｏｎ）（８バイト）

意味：コンテンツ毎の暗号処理用のキーであルコンテンツキーをメモリカードのストレージキー（Ｋｓｔｒ）で暗号化したデータ。

機能：コンテンツの暗号処理に使用される

値：０から０ｘＦＦＦＦＦＦＦＦＦＦＦＦＦＦＦＦまで

【０１２３】

Ｃ＿ＭＡＣ［ｎ］（８バイト）

意味：著作権情報改ざんチェック値

機能：コンテンツ累積番号を含む複数のＴＲＫＩＮＦの内容と隠しシーケンス番号から作成される値。隠しシーケンス番号とは、メモリカードの隠し領域に記録されているシーケンス番号のことである。著作権対応でないレコーダは、隠し領域を読むことができない。また、著作権対応の専用のレコーダ、またはメモリカードを読むことを可能とするアプリケーションを搭載したパーソナルコンピュータは、隠し領域をアクセスすることができる。

【０１２４】

Ａ（１バイト）

意味：パーツの属性

機能：パーツ内の圧縮モード等の情報を示す

値：図２３を参照して以下に説明する

ただし、Ｎ＝０，１のモノラルは、ｂｉｔ７が１でサブ信号を０、メイン信号（Ｌ＋Ｒ）のみの特別なＪｏｉｎｔモードをモノラルとして規定する。ｂｉｔ２，１の情報は通常の再生機は無視しても構わない。

【０１２５】

Ａのビット０は、エンファシスのオン／オフの情報を形成し、ビット１は、再生ＳＫＩＰか、通常再生かの情報を形成し、ビット２は、データ区分、例えばオーディオデータか、ＦＡＸ等の他のデータかの情報を形成する。ビット３は、未定義である。ビット４、５、６を組み合わせることによって、図示のように、ＡＴＲＡＣ３のモード情報が規定される。すなわち、Ｎは、この３ビットで表されるモードの値であり、モノ（Ｎ＝０，１），ＬＰ（Ｎ＝２），ＳＰ（Ｎ＝４），ＥＸ（Ｎ＝５），ＨＱ（Ｎ＝７）の５種類のモードについて、記録時間（６４ＭＢのメモリカードの場合）、データ転送レート、１ブロック内のＳＵ数がそれぞれ示されている。１ＳＵのバイト数は、（モノ：１３６バイト、ＬＰ：１９２バイト、ＳＰ：３０４バイト、ＥＸ：３８４バイト、ＨＱ：５１２バイト）である。さらに、ビット７によって、ＡＴＲＡＣ３のモード（０：Dual　１：Joint ）が示される。

【０１２６】

一例として、６４ＭＢのメモリカードを使用し、ＳＰモードの場合について説明する。６４ＭＢのメモリカードには、３９６８ブロックがある。ＳＰモードでは、１ＳＵが３０４バイトであるので、１ブロックに５３ＳＵが存在する。１ＳＵは、（１０２４／４４１００）秒に相当する。従って、１ブロックは、（１０２４／４４１００）×５３×（３９６８－１６）＝４８６３秒＝８１分

転送レートは、

（４４１００／１０２４）×３０４×８＝１０４７３７ bps

となる。

【０１２７】

ＬＴ（１バイト）

意味：再生制限フラグ（ビット７およびビット６）とセキュリティバージョン（ビット５－ビット０）

機能：このトラックに関して制限事項があることを表す

値：ビット７：　０＝制限なし　　１＝制限有り

　　ビット６：　０＝期限内　　１＝期限切れ

ビット５－ビット０：セキュリティバージョン０（０以外であれば再生禁止とする）

【０１２８】

ＦＮｏ（２バイト）

意味：ファイル番号

機能：最初に記録された時のトラック番号、且つこの値は、メモリカード内の隠し領域に記録されたＭＡＣ計算用の値の位置を特定する

値：１から０ｘ１９０（４００）

【０１２９】

ＭＧ（Ｄ）ＳＥＲＩＡＬ－ｎｎｎ（１６バイト（ｕｐｐｅｒ：８，Ｌｏｗｅｒ：８））

意味：記録機器のセキュリティブロック（セキュリティＩＣ２０）のシリアル番号

機能：記録機器ごとに全て異なる固有の値

値：０から0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

【０１３０】

ＣＯＮＮＵＭ（４バイト）

意味：コンテンツ累積番号

機能：曲毎に累積されていく固有の値で記録機器のセキュリティブロックによって管理される。２の３２乗、４２億曲分用意されており、記録した曲の識別に使用する

値：０から０ｘＦＦＦＦＦＦＦＦ。

【０１３１】

ＹＭＤｈｍｓ－Ｓ（４バイト）（Ｏｐｔｉｏｎ）

意味：再生制限付きのトラックの再生開始日時

機能：ＥＭＤで指定する再生開始を許可する日時

値：上述した日時の表記と同じ

ＹＭＤｈｍｓ－Ｅ（４バイト）（Ｏｐｔｉｏｎ）

意味：再生制限付きのトラックの再生終了日時

機能：ＥＭＤで指定する再生許可を終了する日時

値：上述した日時の表記と同じ

【０１３２】

ＭＴ（１バイト）（Ｏｐｔｉｏｎ）

意味：再生許可回数の最大値

機能：ＥＭＤで指定される最大の再生回数

値：１から０ｘＦＦ　未使用の時は、０ｘ００

ＬＴのｂｉｔ７の値が０の場合はＭＴの値は００とすること

【０１３３】

ＣＴ（１バイト）（Ｏｐｔｉｏｎ）

意味：再生回数

機能：再生許可された回数の内で、実際に再生できる回数。再生の度にデクリメントする

値：０ｘ００～０ｘＦＦ　未使用の時は、０ｘ００である

ＬＴのｂｉｔ７が１でＣＴの値が００の場合は再生を禁止すること。

【０１３４】

ＣＣ（１バイト）

意味：COPY CONTROL

機能：コピー制御

値：図２４に示すように、ビット６および７によってコピー制御情報を表し、ビット４および５によって高速ディジタルコピーに関するコピー制御情報を表し、ビット２および３によってセキュリティブロック認証レベルを表す。ビット０および１は、未定義

ＣＣの例：（ｂｉｔ７，６）１１：無制限のコピーを許可、０１：コピー禁止、００：１回のコピーを許可（ｂｉｔ３，２）００：アナログないしディジタルインからの録音、ＭＧ認証レベルは０とする

ＣＤからのディジタル録音では（ｂｉｔ７，６）は００、（ｂｉｔ３，２）は００となる

【０１３５】

ＣＮ（１バイト）（Ｏｐｔｉｏｎ）

意味：高速ディジタルコピーＨＳＣＭＳ(High speed Serial Copy Management System)におけるコピー許可回数

機能：コピー１回か、コピーフリーかの区別を拡張し、回数で指定する。コピー第１世代の場合にのみ有効であり、コピーごとに減算する

値：００：コピー禁止、０１から０ｘＦＥ：回数、０ｘＦＦ：回数無制限。

【０１３６】

上述したトラック情報領域ＴＲＫＩＮＦに続いて、０ｘ０３７０から始まる２４バイトのデータをパーツ管理用のパーツ情報領域ＰＲＴＩＮＦと呼び、１つのトラックを複数のパーツで構成する場合に、時間軸の順番にＰＲＴＩＮＦを並べていく。図２５にＰＲＴＩＮＦの部分を示す。ＰＲＴＩＮＦ内のデータについて、配置順序に従って以下に説明する。

【０１３７】

ＰＲＴＳＩＺＥ（４バイト）

意味：パーツサイズ

機能：パーツの大きさを表す。クラスタ：２バイト（最上位）、開始ＳＵ：１バイト（上位）、終了ＳＵ：１バイト（最下位）

値：クラスタ：１から０ｘ１Ｆ４０（８０００）、開始ＳＵ：０から０ｘＡ０（１６０）、終了ＳＵ：０から０ｘＡ０（１６０）（但し、ＳＵの数え方は、０，１，２，と０から開始する）

【０１３８】

ＰＲＴＫＥＹ（８バイト）

意味：パーツを暗号化するための値

機能：初期値＝０、編集時は編集の規則に従うこと

値：０から０ｘＦＦＦＦＦＦＦＦＦＦＦＦＦＦＦＦ

【０１３９】

ＣＯＮＮＵＭ０（４バイト）

意味：最初に作られたコンテンツ累積番号キー

機能：コンテンツをユニークにするためのＩＤの役割

値：コンテンツ累積番号初期値キーと同じ値とされる。

【０１４０】

図２０に戻る。ＡＴＲＡＣ３データファイルの属性ヘッダ中には、図２０に示すように、付加情報ＩＮＦが含まれる。ＩＮＦは、トラックに関する付加情報データであり、ヘッダを伴った可変長の付加情報データ。複数の異なる付加情報が並べられることがある。それぞれにＩＤとデータサイズが付加されている。個々のヘッダを含む付加情報データは、最小１６バイト以上で４バイトの整数倍の単位である。

【０１４１】

上述した属性ヘッダに対して、ＡＴＲＡＣ３データファイルの各ブロックのデータが続く。図２６に示すように、ブロック毎にヘッダが付加される。各ブロックのデータについて以下に説明する。

【０１４２】

ＢＬＫＩＤ－Ａ３Ｄ（４バイト）

意味：BLOCKID FILE ID

機能：ＡＴＲＡＣ３データの先頭であることを識別するための値

値：固定値＝”Ａ３Ｄ”（例えば０ｘ４１３３４４２０）

【０１４３】

ＭＣｏｄｅ（２バイト）

意味：MAKER CODE

機能：記録した機器の、メーカー、モデルを識別するコード

値：上位１０ビット（メーカーコード）　下位６ビット（機種コード）

【０１４４】

ＣＯＮＮＵＭ０（４バイト）

意味：最初に作られたコンテンツ累積番号

機能：コンテンツをユニークにするためのＩＤの役割、編集されても値は変化させない

値：コンテンツ累積番号初期値キーと同じ値とされる

【０１４５】

ＢＬＯＣＫ　ＳＥＲＩＡＬ（４バイト）

意味：トラック毎に付けられた連続番号

機能：ブロックの先頭は０から始まり次のブロックは＋１づつインクリメント編集されても値を変化させない

値：０より始まり０ｘＦＦＦＦＦＦＦＦまで

【０１４６】

ＢＬＯＣＫ－ＳＥＥＤ（８バイト）

意味：１ブロックを暗号化するための１つの鍵

機能：ブロックの先頭は、記録機器のセキュリティブロックで乱数を生成、続くブロックは、＋１インクリメントされた値、この値が失われると、１ブロックに相当する約１秒間、音が出せないために、ヘッダとブロック末尾に同じものが二重に書かれる。編集されても値を変化させない

値：初期は８バイトの乱数

【０１４７】

ＩＮＩＴＩＡＬＩＺＡＴＩＯＮ　ＶＥＣＴＯＲ（８バイト）

意味：ブロック毎にＡＴＲＡＣ３データを暗号化、復号化する時に必要な初期値

機能：ブロックの先頭は０から始まり、次のブロックは最後のＳＵの最後の暗号化された８バイトの値。デバイドされたブロックの途中からの場合は開始ＳＵの直前の最後の８バイトを用いる。編集されても値を変化させない

値：０から０ｘＦＦＦＦＦＦＦＦＦＦＦＦＦＦＦＦ

【０１４８】

ＳＵ－ｎｎｎ

意味：サウンドユニットのデータ

機能：１０２４サンプルから圧縮されたデータ、圧縮モードにより出力されるバイト数が異なる。編集されても値を変化させない（一例として、ＳＰモードの時では、Ｎ＝３８４バイト）

値：ＡＴＲＡＣ３のデータ値。

【０１４９】

図２０では、Ｎ＝３８４であるので、１ブロックに４２ＳＵが書かれる。また、１ブロックの先頭の２つのスロット（４バイト）がヘッダとされ、最後の１スロット（２バイト）にＢＬＫＩＤ－Ａ３Ｄ、ＭＣｏｄｅ、ＣＯＮＮＵＭ０、ＢＬＯＣＫ　ＳＥＲＩＡＬが二重に書かれる。従って、１ブロックの余りの領域Ｍバイトは、（１６，３８４－３８４×４２－１６×３＝２０８（バイト）となる。この中に上述したように、８バイトのＢＬＯＣＫ　ＳＥＥＤが二重に記録される。

【０１５０】

ここで、フラッシュメモリ３４に記憶されているデータは、後述するように例えば、ＡＴＲＡＣ３方式で圧縮されている。圧縮の単位がサウンドユニットＳＵである。従って、記憶装置３００から再生装置２００にデータを読み出す場合には、読み出しの最小単位は当該サウンドユニットＳＵとなる。オーディオデータの圧縮方式は、ＡＴＲＡＣ３などのＡＴＲＡＣ方式以外のＣＯＤＥＣ方式でもよい。

【０１５１】

ブロックシードデータＢＳは、各ブロック毎に例えば乱数を発生して生成されたデータである。

【０１５２】

〔フラッシュメモリ管理モジュール３５〕

フラッシュメモリ管理モジュール３５は、フラッシュメモリ３４へのデータの書き込み、フラッシュメモリ３４からのデータの読み出しなどの制御を行う。

【０１５３】

図１５に示す再生装置２００の構成について説明する。再生装置２００は、例えば、主制御モジュール４１、通信インターフェイス４２、制御モジュール４３、編集モジュール４４、圧縮／伸長モジュール４５、スピーカ４６、Ｄ／Ａ変換器４７およびＡ／Ｄ変換器４８を有する。

【０１５４】

〔主制御モジュール４１〕

主制御モジュール４１は、再生装置２００の処理を統括的に制御する。

【０１５５】

〔制御モジュール４３〕

図１５に示すように、制御モジュール４３は、例えば、乱数発生ユニット６０、記憶ユニット６１、鍵生成／鍵演算ユニット６２、相互認証ユニット６３、暗号化／復号ユニット６４および制御ユニット６５を有する。制御モジュール４３は、制御モジュール３３と同様に、シングルチップの暗号処理専用の集積回路であり、多層構造を有し、内部のメモリセルはアルミニウム層などのダミー層に挟まれている。また、制御モジュール４３は、動作電圧または動作周波数の幅が狭く、外部から不正にデータを読み出せないように耐タンパー性を有している。乱数発生ユニット６０は、乱数発生指示を受けると、６４ビット（８バイト）の乱数を発生する。記憶ユニット６１は、認証処理に必要な種々のデータを記憶している。

【０１５６】

鍵生成／鍵演算ユニット６２は、例えば、ＩＳＯ／ＩＥＣ９７９７のＭＡＣ演算方式を用いた演算などの種々の演算を行って鍵データを生成する。このとき、"Block cipher Algorithm"としてＦＩＰＳ ＰＵＢ ４６－２に規定されるＤＥＳが用いられる。

【０１５７】

相互認証ユニット６３は、例えば、コンピュータから入力したオーディオデータを記憶装置３００に出力する動作を行うのに先立って、記憶装置３００との間で相互認証処理を行う。また、相互認証ユニット６３は、記憶装置３００からオーディオデータを入力する動作を行うのに先立って、記憶装置３００との間で相互認証処理を行う。また、相互認証ユニット６３は、相互認証処理において、前述したＭＡＣ演算を行う。当該相互認証処理では、記憶ユニット６１に記憶されているデータが用いられる。なお、相互認証ユニット６３は、必要に応じて、例えば、パーソナルコンピュータ（ＰＣ）１００あるいはネットワーク上のコンピュータとの間でオーディオデータの入出力を行う動作に先立って、パーソナルコンピュータ（ＰＣ）１００あるいはネットワーク上のコンピュータとの間で相互認証処理を行う。

【０１５８】

暗号化／復号ユニット６４は、前述したように、ＦＩＰＳ ＰＵＢ ８１に規定されたＥＣＢモードおよびＣＢＣモードを選択的に用いてブロック暗号化を行う。

【０１５９】

暗号化／復号ユニット６４は、ＦＩＰＳ８１のモードのうち、ＥＣＢモードおよびＣＢＣモードの復号を選択的に行う。ここで、暗号化／復号ユニット６４は、ＣＢＣモードにおいて、例えば５６ビットの鍵データｋを用いて、暗号文を、６４ビットからなる暗号化ブロックを単位として復号して平文を生成する。

【０１６０】

制御ユニット６５は、乱数発生ユニット６０、記憶ユニット６１、鍵生成／鍵演算ユニット６２、相互認証ユニット６３および暗号化／復号ユニット６４の処理を統括的に制御する。

【０１６１】

〔編集モジュール４４〕

編集モジュール４４は、例えば、図１６に示すように記憶装置３００のフラッシュメモリ３４内に記憶されたトラックデータファイルを、ユーザからの操作指示に基づいて編集して新たなトラックデータファイルを生成する。

【０１６２】

〔圧縮／伸長モジュール４５〕

圧縮／伸長モジュール４５は、例えば、記憶装置３００から入力した暗号化されたオーディオデータを復号した後に再生する際に、ＡＴＲＡＣ３方式で圧縮されているオーディオデータを伸長し、当該伸長したオーディオデータをＤ／Ａ変換器４７に出力する。また、例えば、ＣＤ、ＤＶＤあるいはＰＣ１から入力したオーディオデータを、記憶装置３００に記憶する際に、当該オーディオデータをＡＴＲＡＣ３方式で圧縮する。

【０１６３】

〔Ｄ／Ａ変換器４７〕

Ｄ／Ａ変換器４７は、圧縮／伸長モジュール４５から入力したデジタル形式の

オーディオデータをアナログ形式のオーディオデータに変換してスピーカ４６に出力する。

【０１６４】

〔スピーカ４６〕

スピーカ４６は、Ｄ／Ａ変換器４７から入力したオーディオデータに応じた音響を出力する。

【０１６５】

〔Ａ／Ｄ変換器４８〕

Ａ／Ｄ変換器４８は、例えば、ＣＤプレーヤ７から入力したアナログ形式のオーディオデータをデジタル形式に変換して圧縮／伸長モジュール４５に出力する。

【０１６６】

［コンテンツデータの記憶装置に対する格納処理および再生処理］

図１５に示す再生装置２００と、記憶装置３００との間では、コンテンツデータの移動、すなわち、再生装置２００から記憶装置３００のフラッシュメモリ３４に対するデータ記録処理が実行され、さらに、記憶装置３００のフラッシュメモリ３４から再生装置２００に対するデータ再生処理が実行される。

【０１６７】

このデータ記録および再生処理について、以下説明する。まず、再生装置２００から記憶装置３００のフラッシュメモリ３４に対するデータ記録処理を図２７のフローを用いて説明する。

【０１６８】

再生装置および記憶装置は、データ移動に先立ち、まずステップＳ２７０１、Ｓ２７０２に示す相互認証処理を実行する。図２８に、共通鍵暗号方式を用いた相互認証方法（ISO/IEC 9798-2）を示す。図２８においては、共通鍵暗号方式としてＤＥＳを用いているが、共通鍵暗号方式であれば他の方式も可能である。図２８において、まず、Ｂが６４ビットの乱数Ｒｂを生成し、Ｒｂおよび自己のＩＤであるＩＤ（ｂ）をＡに送信する。これを受信したＡは、新たに６４ビットの乱数Ｒａを生成し、Ｒａ、Ｒｂ、ＩＤ（ｂ）の順に、ＤＥＳのＣＢＣモードで鍵

Ｋａｂを用いてデータを暗号化し、Ｂに返送する。なお、鍵Ｋａｂは、ＡおよびＢに共通の秘密鍵としてそれぞれの記録素子内に格納する鍵である。ＤＥＳのＣＢＣモードを用いた鍵Ｋａｂによる暗号化処理は、例えばＤＥＳを用いた処理においては、初期値とＲａとを排他的論理和し、ＤＥＳ暗号化部において、鍵Ｋａｂを用いて暗号化し、暗号文Ｅ１を生成し、続けて暗号文Ｅ１とＲｂとを排他的論理和し、ＤＥＳ暗号化部において、鍵Ｋａｂを用いて暗号化し、暗号文Ｅ２を生成し、さらに、暗号文Ｅ２とＩＤ（ｂ）とを排他的論理和し、ＤＥＳ暗号化部において、鍵Ｋａｂを用いて暗号化して生成した暗号文Ｅ３とによって送信データ（Token-AB）を生成する。

【０１６９】

これを受信したＢは、受信データを、やはり共通の秘密鍵としてそれぞれの記録素子内に格納する鍵Ｋａｂ（認証キー）で復号化する。受信データの復号化方法は、まず、暗号文Ｅ１を認証キーＫａｂで復号化し、乱数Ｒａを得る。次に、暗号文Ｅ２を認証キーＫａｂで復号化し、その結果とＥ１を排他的論理和し、Ｒｂを得る。最後に、暗号文Ｅ３を認証キーＫａｂで復号化し、その結果とＥ２を排他的論理和し、ＩＤ(ｂ)を得る。こうして得られたＲａ、Ｒｂ、ＩＤ(ｂ)のうち、ＲｂおよびＩＤ(ｂ)が、Ｂが送信したものと一致するか検証する。この検証に通った場合、ＢはＡを正当なものとして認証する。

【０１７０】

次にＢは、認証後に使用するセッションキー（Ｋｓｅｓ）を生成する（生成方法は、乱数を用いる）。そして、Ｒｂ、Ｒａ、Ｋｓｅｓの順に、ＤＥＳのＣＢＣモードで認証キーＫａｂを用いて暗号化し、Ａに返送する。

【０１７１】

これを受信したＡは、受信データを認証キーＫａｂで復号化する。受信データの復号化方法は、Ｂの復号化処理と同様であるので、ここでは詳細を省略する。こうして得られたＲｂ、Ｒａ、Ｋｓｅｓの内、ＲｂおよびＲａが、Ａが送信したものと一致するか検証する。この検証に通った場合、ＡはＢを正当なものとして認証する。互いに相手を認証した後には、セッションキーＫｓｅｓは、認証後の秘密通信のための共通鍵として利用される。

【０１７２】

なお、受信データの検証の際に、不正、不一致が見つかった場合には、相互認証が失敗したものとして処理を終了（Ｓ２７０３でＮｏ）する。

【０１７３】

相互認証が成立（Ｓ２７０３でＹｅｓ）した場合は、ステップＳ２７０４において、再生装置がコンテンツキーＫｃｏｎの生成処理を実行する。この処理は、図１５の乱数生成ユニット６０で生成した乱数を用いて鍵生成／鍵演算ユニット６２において実行される。

【０１７４】

次に、ステップＳ２７０５において、（１）コンテンツキーＫｃｏｎを有効化キーブロック（ＥＫＢ）から取得される暗号化キーＫＥＫを用いて暗号化処理して、Ｅ（ＫＥＫ，Ｋｃｏｎ）を生成するとともに、（２）コンテンツキーＫｃｏｎを認証処理において生成したセッションキー（Ｋｓｅｓ）で暗号化処理を実行して、Ｅ（Ｋｓｅｓ，Ｋｃｏｎ）を生成して、記憶装置（メモリカード）に送信する。

【０１７５】

ステップＳ２７０６では、記憶装置が再生装置から受信したＥ（Ｋｓｅｓ，Ｋｃｏｎ）をセッションキーで復号してコンテンツキーＫｃｏｎを取得し、さらに、Ｋｃｏｎを記憶装置に予め格納されているストレージキーＫｓｔｒによって暗号化してＥ（Ｋｓｔｒ，Ｋｃｏｎ）を生成し、これを再生装置に送信する。

【０１７６】

次に、再生装置は、ステップＳ２７０７において、ステップＳ２７０５で生成したＥ（ＫＥＫ，Ｋｃｏｎ）、およびステップＳ２７０６で記憶装置から受信したＥ（Ｋｓｔｒ，Ｋｃｏｎ）を用いて、データファイル（図２０参照）を構成するトラック情報領域ＴＲＫＩＮＦデータを生成し、データファイルのフォーマット処理の後、これを記憶装置（メモリカード）に送信する。

【０１７７】

ステップＳ２７０８において、記憶装置（メモリカード）は、再生装置から受信したデータファイルをフラッシュメモリに格納する。

【０１７８】

このような処理により、データファイルのトラック情報領域ＴＲＫＩＮＦデータには、先に説明した図２０、図２２に示すように、コンテンツキーＫｃｏｎを有効化キーブロック（ＥＫＢ）から取得される暗号化キーＫＥＫを用いて暗号化処理したＥ（ＫＥＫ，Ｋｃｏｎ）と、コンテンツキーＫｃｏｎを記憶装置に予め格納されているストレージキーＫｓｔｒによって暗号化したＥ（Ｋｓｔｒ，Ｋｃｏｎ）の２つの暗号化コンテンツキーが格納されることになる。

【０１７９】

なお、音楽データ、画像データ等の暗号化処理は、コンテンツキーＫｃｏｎをそのままコンテンツの暗号化鍵として適用して実行するか、あるいはコンテンツを構成するパーツ、またはブロック等を単位として、コンテンツキーと他のキー生成データに基づいて各パーツ単位、またはブロック単位の暗号化鍵を個別に生成して各パーツ単位、またはブロック単位の暗号化処理を行なう構成とすることが可能である。

【０１８０】

このようなデータファイルを用いた再生処理においては、再生装置は、Ｅ（ＫＥＫ，Ｋｃｏｎ）と、Ｅ（Ｋｓｔｒ，Ｋｃｏｎ）のいずれかを選択的に適用してコンテンツキーＫｃｏｎを取得可能となる。

【０１８１】

次に、再生装置２００が記憶装置３００のフラッシュメモリ３４に格納されたデータの読み出し処理、すなわち再生処理を実行する場合の処理を図２９のフローを用いて説明する。

【０１８２】

再生装置および記憶装置は、データ移動に先立ち、まずステップＳ２９０１、Ｓ２９０２に示す相互認証処理を実行する。この処理は、先に説明した図２８の処理と同様である。相互認証が失敗した場合（Ｓ２９０３でＮｏ）は、処理を終了する。

【０１８３】

相互認証が成立（Ｓ２９０３でＹｅｓ）した場合は、ステップＳ２９０４にお

いて、記憶装置が再生装置に対してデータファイルを送信する。データファイルを受信した再生装置は、データファイル中のトラック情報領域ＴＲＫＩＮＦデータを検査し、コンテンツキー（Ｋｃｏｎ）の格納状況を判別する。この判別処理は、キー有効化ブロック（ＥＫＢ）によって取得される暗号化キーＫＥＫによって暗号化されたコンテンツキー、すなわちＥ（ＫＥＫ．Ｋｃｏｎ）が格納されているか否かを判別する処理である。Ｅ（ＫＥＫ．Ｋｃｏｎ）の有無は、先の図２０，２２で説明したデータファイル中のトラック情報領域ＴＲＫＩＮＦデータの［ＥＫＩ］のデータにより判別可能である。

【０１８４】

Ｅ（ＫＥＫ．Ｋｃｏｎ）が格納されている場合（ステップＳ２９０６でＹｅｓ）は、ステップＳ２９０７に進み、キー有効化ブロック（ＥＫＢ）の処理により、暗号化キーＫＥＫを取得して、取得した暗号化キーＫＥＫにより、Ｅ（ＫＥＫ．Ｋｃｏｎ）を復号して、コンテンツキーＫｃｏｎを取得する。

【０１８５】

Ｅ（ＫＥＫ．Ｋｃｏｎ）が格納されていない場合（ステップＳ２９０６でＮｏ）は、ステップＳ２９０８において、記憶装置の制御モジュール３３において、記憶装置に予め格納されているストレージキーＫｓｔｒによって暗号化したＥ（Ｋｓｔｒ，Ｋｃｏｎ）をストレージキーＫｓｔｒによって復号して、さらに、相互認証処理において再生装置および記憶装置で共有したセッションキーＫｓｅｓで暗号化したデータＥ（Ｋｓｅｓ，Ｋｃｏｎ）を生成して、再生装置に送信する。

【０１８６】

再生装置は、ステップＳ２９０９において、記憶装置から受信したＥ（Ｋｓｅｓ，Ｋｃｏｎ）をセッションキーＫｓｅｓで復号してコンテンツキーＫｃｏｎを取得する。

【０１８７】

ステップＳ２９１０では、ステップＳ２９０７、またはステップＳ２９０９のいずれかにおいて取得したコンテンツキーＫｃｏｎにより暗号化コンテンツの復号を行なう。

【０１８８】

このように、暗号化コンテンツの再生処理において、再生装置は、Ｅ（ＫＥＫ，Ｋｃｏｎ）を有効化キーブロック（ＥＫＢ）から取得される暗号化キーＫＥＫを用いて復号するか、または、記憶装置に予め格納されているストレージキーＫｓｔｒによって暗号化したＥ（Ｋｓｔｒ，Ｋｃｏｎ）に基づく処理を実行するか、いずれかの処理を実行することによりコンテンツキーＫｃｏｎを取得することができる。

【０１８９】

なお、音楽データ、画像データ等の復号処理は、コンテンツキーＫｃｏｎをそのままコンテンツの復号鍵として適用して実行するか、あるいはコンテンツを構成するパーツ、またはブロック等を単位として、コンテンツキーと他のキー生成データに基づいて各パーツ単位、またはブロック単位の復号鍵を個別に生成して各パーツ単位、またはブロック単位の復号処理を行なう構成とすることが可能である。

【０１９０】

［ＫＥＫを格納したＥＫＢのフォーマット］

先に図６を用いて有効化キーブロック（ＥＫＢ）の概略的なフォーマットについて説明したが、さらに、キー暗号化キー（ＫＥＫ）を有効化キーブロック（ＥＫＢ）に格納して保持する場合の具体的なデータ構成例について説明する。

【０１９１】

図３０にキー暗号化キー（ＫＥＫ）を有効化キーブロック（ＥＫＢ）に格納したデータであるＥＫＢである配信鍵許可情報ファイルの構成例を示す。デバイス（再生装置）は、このファイルから必要に応じてキー暗号化キー（ＫＥＫ）を取り出して、ＫＥＫによりＥ（ＫＥＫ，Ｋｃｏｎ）を復号してコンテンツキー：Ｋｃｏｎを取得してコンテンツの復号を実行する。各データについて説明する。

【０１９２】

ＢＬＫＩＤ－ＥＫＢ（４バイト）

意味：BLOCKID FILE ID

機能：配信鍵情報ファイルの先頭であることを識別するための値

値：固定値＝”ＥＫＢ”（例えば０ｘ４５４Ｂ４２２０）

【０１９３】

ＭＣｏｄｅ（２バイト）

意味：MAKER CODE

機能：記録した機器の、メーカー、モデルを識別するコード

値：上位１０ビット（メーカーコード）　下位６ビット（機種コード）

【０１９４】

ＬＫＦ

意味：LINK FILE INFORMATION

機能：このＥＫＢによって取得されるＫＥＫが適用可能なコンテンツデータであるリンクファイルを識別する。

値：０～０ｘＦＦ

ｂｉｔ７：再生管理ファイル（ＰＢＬＩＳＴ）に使用：１、未使用：０

ｂｉｔ６：改竄チェック値（ＩＣＶ）に使用：１、未使用：０

ｂｉｔ５～０：リザーブ

【０１９５】

ＬＩＮＫ　ｃｏｕｎｔ

意味：LINK COUNT

機能：リンクしているファイル（例えばＡＴＲＡＣＫ３ファイル）数

値：０～０ｘＦＦＦＦＦＦＦＦ

【０１９６】

Ｖｅｒｓｉｏｎ

意味：VERSION

機能：配信鍵許可情報ファイルのバージョンを示す。

値：０～０ｘＦＦＦＦＦＦＦＦ

【０１９７】

ＥＡ

意味：Encryption Algorithm

機能：配信鍵許可情報ファイルのトレース処理アルゴリズムを示す。

値：０～０ｘＦＦ

００ｈ：３ＤＥＳ：トリプルＤＥＳモードによる処理

０１ｈ：ＤＥＳ　：シングルＤＥＳモードによる処理

なお、トリプルＤＥＳモードによる処理は、２種類以上の暗号処理キーを用いる暗号処理であり、シングルＤＥＳモードは１つのキーによる処理である。

【０１９８】

ＫＥＫ１

意味：Key Encrypting Key

機能：キー有効化ブロック（ＥＫＢ）中のルートキー（最上位）キーで暗号化されたコンテンツキー暗号キー

値：０～０ｘＦＦＦＦＦＦＦＦＦＦＦＦＦＦＦＦ

【０１９９】

ＫＥＫ２

意味：Key Encrypting Key

機能：キー有効化ブロック（ＥＫＢ）中のルートキー（最上位）キーで暗号化されたコンテンツキー暗号キー

値：０～０ｘＦＦＦＦＦＦＦＦＦＦＦＦＦＦＦＦ

【０２００】

Ｅ（Ｖｅｒｓｉｏｎ）

意味：Encrypted Version

機能：キー有効化ブロック（ＥＫＢ）中のルートキー（最上位）キーで暗号化されたバージョン番号。復号時の下４バイトはリザーブ

値：０～０ｘＦＦＦＦＦＦＦＦＦＦＦＦＦＦＦＦ

【０２０１】

Ｓｉｚｅ ｏｆ ｔａｇ ｐａｒｔ

意味：Size of tag part

機能：配信鍵許可情報ファイルを構成するデータのタグ部分のサイズ（Ｂｙｔｅ）

値：０～０ｘＦＦＦＦＦＦＦＦ

【０２０２】

Ｓｉｚｅ ｏｆ Ｋｅｙ ｐａｒｔ

意味：Size of key part

機能：配信鍵許可情報ファイルを構成するデータのキー部分のサイズ（Ｂｙｔｅ）

値：０～０ｘＦＦＦＦＦＦＦＦ

【０２０３】

Ｓｉｚｅ ｏｆ Ｓｉｇｎ ｐａｒｔ

意味：Size of sign part

機能：配信鍵許可情報ファイルを構成するデータのサイン部分のサイズ（Ｂｙｔｅ）

値：０～０ｘＦＦＦＦＦＦＦＦ

【０２０４】

Ｔａｇ ｐａｒｔ

意味：Tag part

機能：配信鍵許可情報ファイルを構成するデータのタグ部分のデータ

値：すべての値

８バイトに満たない場合は０で埋めて８バイトにする。

【０２０５】

Ｋｅｙ ｐａｒｔ

意味：Key part

機能：配信鍵許可情報ファイルを構成するデータのキー部分のデータ

値：すべての値

【０２０６】

Ｓｉｇｎａｔｕｒｅ ｐａｒｔ

意味：Signature part

機能：配信鍵許可情報ファイルを構成するデータの署名（Ｓｉｇｎａｔｕｒｅ）部分のデータ

値：すべての値

【０２０７】

上述の説明および図３０によって示されるように、デバイスに対して提供される配信鍵許可情報ファイルには、その配信鍵許可情報ファイルから取得されるＫＥＫが適用可能なコンテンツデータであるリンクファイルを識別するための識別データ［ＬＫＦ］が格納され、さらに、リンクしているファイル（例えばＡＴＲＡＣＫ３ファイル）数としてのデータ［Ｌｉｎｃ　Ｃｏｕｎｔ］が格納される。再生装置は、［ＬＫＦ］、［Ｌｉｎｋ　Ｃｏｕｎｔ］を参照することにより、その配信鍵許可情報ファイルから取得されるＫＥＫを適用すべきデータが存在するか否かおよびその数を知ることが可能となる。

【０２０８】

［リンク情報を用いたデータ復号、再生処理］

上述した配信鍵許可情報ファイルに含まれるリンクファイルを識別するための識別データ［ＬＫＦ］、リンクしているファイル（例えばＡＴＲＡＣＫ３ファイル）数としてのデータ［Ｌｉｎｃ　Ｃｏｕｎｔ］を用いて、効率的にデータの復号、再生を実行する処理態様について、以下説明する。

【０２０９】

図３１に記憶装置のデータ格納領域、例えば図１５に示す記憶装置３００のフラッシュメモリ３４に格納されたデータファイル構成例を示す。ここでは、音楽データ（ＨＩＦＩ）のディレクトリ構成のみを例として示しているが、さらに、画像ファイル等のディレクトリが存在してもよい。

【０２１０】

図３１に示す音楽データのディレクトリには、再生管理ファイル（ＰＢＬＩＳＴ）、暗号化コンテンツとして複数のＡＴＲＡＣＫ３データファイル（Ａ３Ｄ）含まれる。さらに、記憶装置には、複数の有効化キーブロックファイル（ＥＫＢｎ）が格納される。ＡＴＲＡＣＫ３データファイル（Ａ３Ｄ）の復号処理に適用するコンテンツキーを取得するための有効化キーブロックファイル（ＥＫＢｎ）は、ＡＴＲＡＣＫ３データファイル（Ａ３Ｄ）に含まれるポインタによって判別される。図３１に示すように、１つの有効化キーブロックファイル（ＥＫＢ１）３１０１は複数（３）のＡＴＲＡＣＫ３データファイル（Ａ３Ｄ）の復号処理に

適用される。

【０２１１】

この場合、有効化キーブロックファイル（ＥＫＢ１）３１０１に対応する配信鍵許可情報ファイルの［Ｌｉｎｃ　Ｃｏｕｎｔ］には３つのコンテンツに適用されることを示すデータが格納されることになる。

【０２１２】

図３１のような複数のコンテンツファイル、複数の有効化キーブロックファイルを格納した記憶装置であるメモリカードからコンテンツを復号して、再生する場合の処理フローを図３２に示す。

【０２１３】

図３２の処理は、例えば記憶装置としてのメモリカードを再生装置にセットした際、あるいはメモリカードを装着した再生装置の電源をＯＮした際に再生装置が実行する処理である。

【０２１４】

まず、ステップＳ３２０１において、再生装置は、各々のＥＫＢファイルのトラック情報を読み取り、［Ｌｉｎｃ　Ｃｏｕｎｔ］をチェックする。さらに、［Ｌｉｎｃ　Ｃｏｕｎｔ］のカウント数が多いものから順に、予め定められた個数［ｎ］のＥＫＢファイルを選択する。個数［ｎ］は、再生装置の所定メモリ領域、すなわちキー暗号化キー：ＫＥＫを格納保持する領域に格納可能な個数に相当する個数として設定される。

【０２１５】

次に、ステップＳ３２０２において、選択したＥＫＢの処理により、複数［ｎ］のキー暗号化キー：ＫＥＫを取得し、これらを再生装置の鍵格納領域として設定されたＲＡＭの所定領域に格納する。

【０２１６】

次に、再生装置は、ステップＳ３２０３において、復号、再生するコンテンツを選択する。さらに、ステップＳ３２０４において、その選択コンテンツの復号に適用するＫＥＫがＲＡＭに格納されているか否かを判定し、Ｙｅｓの場合は、ステップＳ３２０５に進み、その対応ＫＥＫに基づいて、Ｅ（ＫＥＫ，Ｋｃｏｎ

）を復号してコンテンツキーを取得して、ステップＳ３２０９で再生、すなわち、取得したコンテンツキーによるデータの復号、再生処理を実行する。

【０２１７】

ステップＳ３２０４において、選択コンテンツの復号に適用するＫＥＫがＲＡＭに格納されていない場合は、ステップＳ３２０６において、ストレージキーで暗号化されたコンテンツキー、すなわち、Ｅ（Ｋｓｔｒ，Ｋｃｏｎ）の有無を判定し、ある場合は、ステップＳ３２０７において、Ｅ（Ｋｓｔｒ，Ｋｃｏｎ）の復号処理によりコンテンツキーを取得して、ステップＳ３２０９で再生、すなわち、取得したコンテンツキーによるデータの復号、再生処理を実行する。

【０２１８】

また、ステップＳ３２０６において、Ｅ（Ｋｓｔｒ，Ｋｃｏｎ）がないと判定されると、その復号対象コンテンツに適用すべきＥＫＢを記憶装置から取得して、取得したＥＫＢの復号処理によりＫＥＫを取得し、取得したＫＥＫによるＥ（ＫＥＫ，Ｋｃｏｎ）の復号処理を実行してコンテンツキーを取得して、ステップＳ３２０９で再生、すなわち、取得したコンテンツキーによるデータの復号、再生処理を実行する。

【０２１９】

このように、再生装置は、予め記憶装置に格納した複数のキー有効化ブロック（ＥＫＢ）の［Ｌｉｎｃ　Ｃｏｕｎｔ］をチェックし、［Ｌｉｎｃ　Ｃｏｕｎｔ］のカウント数が多いＥＫＢの復号を実行して、キー暗号化キー：ＫＥＫを格納しておく構成とすることにより、コンテンツ再生処理の際に、高い確率でＲＡＭに格納したＫＥＫを適用可能となり、効率的なコンテンツ再生が実行できる。

【０２２０】

［キー有効化ブロック（ＥＫＢ）による認証キー配信］

上述の有効化キーブロック（ＥＫＢ）を使用したキーの配信において、認証処理を実行する際に使用する認証キーＩＫｎを配信することにより、安全な秘密鍵として共有する認証キーを提供し、共通鍵方式に従った認証処理を実行する構成について説明する。

【０２２１】

共通鍵暗号方式を用いた相互認証方法（ISO/IEC 9798-2）は、先に図２８を用いて説明した処理であり、データ送受信が実行される前の処理として、双方の正当性を確認するための処理として実行される。認証処理においては、データの送受信を行なう、例えば再生装置と記憶装置は認証キーＫａｂを共有する。この共通鍵Ｋａｂを上述の有効化キーブロック（ＥＫＢ）を使用して再生装置に配信する。

【０２２２】

図３３および図３４に複数のデバイスに共通の認証キーＩＫｎを有効化キーブロック（ＥＫＢ）によって配信する構成例を示す。図３３はデバイス０，１，２，３に対して復号可能な認証キーＩＫｎを配信する例、図３４はデバイス０，１，２，３中のデバイス３をリボーク（排除）してデバイス０，１，２に対してのみ復号可能な認証キーを配信する例を示す。

【０２２３】

図３３の例では、更新ノードキーＫ（ｔ）００によって、認証キーＩＫｎを暗号化したデータ（ｂ）とともに、デバイス０，１，２，３においてそれぞれの有するノードキー、リーフキーを用いて更新されたノードキーＫ（ｔ）００を復号可能な有効化キーブロック（ＥＫＢ）を生成して配信する。それぞれのデバイスは、図３３の右側に示すようにまず、ＥＫＢを処理（復号）することにより、更新されたノードキーＫ（ｔ）００を取得し、次に、取得したノードキーＫ（ｔ）００を用いて暗号化された認証キー：Ｅｎｃ（Ｋ（ｔ）００，ＩＫｎ）を復号して認証キーＩＫｎを得ることが可能となる。

【０２２４】

その他のデバイス４，５，６，７…は同一の有効化キーブロック（ＥＫＢ）を受信しても自身の保有するノードキー、リーフキーでは、ＥＫＢを処理して更新されたノードキーＫ（ｔ）００を取得することができないので、安全に正当なデバイスに対してのみ認証キーを送付することができる。

【０２２５】

一方、図３４の例は、デバイス３が、例えば鍵の漏洩によりリボーク（排除）されているとして、他のグループのメンバ、すなわち、デバイス０，１，２，に

対してのみ復号可能な有効化キーブロック（ＥＫＢ）を生成して配信した例である。図３４に示す（ａ）有効化キーブロック（ＥＫＢ）と、（ｂ）認証キー（ＩＫｎ）をノードキー（Ｋ（ｔ）００）で暗号化したデータを配信する。

【０２２６】

図３４の右側には、復号手順を示してある。デバイス０，１，２は、まず、受領した有効化キーブロックから自身の保有するリーフキーまたはノードキーを用いた復号処理により、更新ノードキー（Ｋ（ｔ）００）を取得する。次に、Ｋ（ｔ）００による復号により認証キーＩＫｎを取得する。

【０２２７】

他のグループのデバイス、例えばデバイス４，５，６…は、この同様のデータ（ＥＫＢ）を受信したとしても、自身の保有するリーフキー、ノードキーを用いて更新ノードキー（Ｋ（ｔ）００）を取得することができない。同様にリボークされたデバイス３においても、自身の保有するリーフキー、ノードキーでは、更新ノードキー（Ｋ（ｔ）００）を取得することができず、正当な権利を有するデバイスのみが認証キーを復号して利用することが可能となる。

【０２２８】

このように、ＥＫＢを利用した認証キーの配送を用いれば、データ量を少なくして、かつ安全に正当権利者のみが復号可能とした認証キーを配信することが可能となる。また、有効化キーブロック（ＥＫＢ）によって暗号化され提供されるＥＫＢ配信認証鍵は、世代（バージョン）管理がなされ、世代毎の更新処理が実行され、任意のタイミングでのデバイスのリボーク（排除）が可能である。

【０２２９】

上述したＥＫＢによる認証キーの提供処理により、リボークされたデバイス（再生装置）では、記憶装置（例えばメモリカード）との認証処理が成立せず、データの不正な復号が不可能となる。

【０２３０】

さらに、ＥＫＢを利用した認証キーの配送を用いれば、メモリカード以外の記憶媒体、例えば再生装置に内蔵したハードディスク等の記憶媒体に対するデータ格納、再生処理に対する制御も可能となる。

【０２３１】

先の図２７～２９を用いて説明したように、記憶装置を利用したコンテンツの記録、再生処理においては、相互認証処理が実行され、相互認証処理の成立を条件として、データの記録および再生が可能となる。この認証処理プログラムは、メモリカードのような相互認証処理が可能な記憶装置との間での処理においては有効に作用するが、例えば、再生装置がハードディスク、ＣＤ－Ｒ等、暗号処理機能を持たない、すなわち相互認証を実行不可能な記憶媒体に対してデータ格納、データ再生時には意味をなさないことになる。シカシ、本発明のシステムでは、このような認証不可能な機器を利用したデータ格納、あるいはデータ再生処理においても認証処理プログラムを実行させる構成とする。ハードディスク、ＣＤ－Ｒ等は相互認証が不可能であるので、仮想のメモリカード（メモリスティック）を再生装置に構成し、仮想メモリカードと再生装置間において認証処理を実行させて、認証成立を条件として、認証機能を持たない記憶媒体に対するデータ格納処理、あるいは記憶媒体からのデータ再生を可能とする。

【０２３２】

これらの仮想メモリカードを使用したデータ記録、再生処理フローを図３５に示す。まず、再生装置は、再生装置内の仮想メモリカードとの間で相互認証処理を実行する。ステップＳ３５０２において、認証成立したか否かを判定し、成立したことを条件としてステップＳ３５０３に進み、認証機能を持たない記憶媒体、例えばハードディスク、ＣＤ－Ｒ、ＤＶＤ等を用いたデータ記録、再生処理を実行する。

【０２３３】

ステップＳ３５０２において、認証が成立しなかったと判定された場合は、ステップＳ３５０３の認証機能を持たない記憶媒体、例えばハードディスク、ＣＤ－Ｒ、ＤＶＤ等を用いたデータ記録、再生処理が実行されない。

【０２３４】

ここで、仮想メモリカードには、予め、先の図１６で説明した認証鍵データを格納した構成とし、再生装置が使用する認証キーを前述したように、キー有効化ブロックで提供する構成とする。

【０２３５】

このように、再生装置の認証キーをキー有効化ブロック（ＥＫＢ）で提供することにより、正当なライセンスを持つデバイス（再生装置）に対してのみ、仮想メモリカードとの相互認証可能な認証キーを配信することが可能となる。従って、不正な機器、すなわちリボークされた再生装置には、有効な認証キーが配信しない処理が可能となる。有効な認証キーが提供されない再生装置は、相互認証が不成立となり、認証機能を持つメモリカードのみならず、認証機能を持たない記憶媒体、例えばハードディスク、ＣＤ－Ｒ、ＤＶＤ等を用いたデータ記録、再生処理が実行されず、不正な機器によるデータ記録、再生を排除することが可能となる。

【０２３６】

すなわち、認証鍵を提供する有効化キーブロック（ＥＫＢ）をキーツリーのリーフを構成するデータ処理装置中、正当なライセンスを持つデータ処理装置においてのみ復号可能で、正当ライセンスを持たない不正なデータ処理装置においては復号不可能な有効化キーブロック（ＥＫＢ）として提供することにより、不正なデータ処理装置における仮想メモリデバイスとの認証成立を防止して、不正データ処理装置におけるコンテンツ利用を排除可能とした構成を有するライセンスシステムが実現される。

【０２３７】

[チェック値（ＩＣＶ:Integrity Check Value）格納構成]

次に、コンテンツの改竄を防止するためにコンテンツのインテグリティ・チェック値（ＩＣＶ）を生成して、コンテンツに対応付けて、ＩＣＶの計算により、コンテンツ改竄の有無を判定する処理構成について説明する。

【０２３８】

コンテンツのインテグリティ・チェック値（ＩＣＶ）は、例えばコンテンツに対するハッシュ関数を用いて計算され、ＩＣＶ＝ｈａｓｈ（Ｋｉｃｖ，Ｃ１，Ｃ２，…）によって計算される。ＫｉｃｖはＩＣＶ生成キーである。Ｃ１，Ｃ２はコンテンツの情報であり、コンテンツの重要情報のメッセージ認証符号（ＭＡＣ:Message authentication Code）が使用される。前述したように、［ＭＡＣ］

は、図２０で説明したＡＴＲＡＣ３データファイルにも含まれる。これらを使用してインテグリティ・チェック値（ＩＣＶ）の計算がなされる。

【０２３９】

ＤＥＳ暗号処理構成を用いたＭＡＣ値生成例を図３６に示す。図３６の構成に示すように対象となるメッセージを８バイト単位に分割し、（以下、分割されたメッセージをＭ１、Ｍ２、・・・、ＭＮとする）、まず、初期値（Initial Value（以下、ＩＶとする））とＭ１を排他的論理和する（その結果をＩ１とする）。次に、Ｉ１をＤＥＳ暗号化部に入れ、鍵（以下、Ｋ１とする）を用いて暗号化する（出力をＥ１とする）。続けて、Ｅ１およびＭ２を排他的論理和し、その出力Ｉ２をＤＥＳ暗号化部へ入れ、鍵Ｋ１を用いて暗号化する（出力Ｅ２）。以下、これを繰り返し、全てのメッセージに対して暗号化処理を施す。最後に出てきたＥＮがメッセージ認証符号（ＭＡＣ（Message Authentication Code））となる。なお、メッセージとしては、検証対象となるコンテンツおよびヘッダ情報等のコンテンツ関連データを構成する部分データが使用可能である。

【０２４０】

このようなコンテンツのＭＡＣ値とＩＣＶ生成キーＫｉｃｖにハッシュ関数を適用して用いてコンテンツのインテグリティ・チェック値（ＩＣＶ）が生成される。改竄のないことが保証された例えばコンテンツ生成時に生成したＩＣＶと、新たにコンテンツに基づいて生成したＩＣＶとを比較して同一のＩＣＶが得られればコンテンツに改竄のないことが保証され、ＩＣＶが異なれば、改竄があったと判定される。

【０２４１】

上述のようなインテグリティ・チェック値（ＩＣＶ）は、コンテンツ個々に対して生成される複数のコンテンツＭＡＣ値により、１つのインテグリティ・チェック値（ＩＣＶ）を生成することが可能である。複数のＭＡＣによるＩＣＶの計算は、例えば、ＩＣＶ＝ＭＡＣ（Ｋｉｃｖ，Ｃ＿ＭＡＣ［０］||Ｃ＿ＭＡＣ［１］||Ｃ＿ＭＡＣ［２］||…）によって生成する。

【０２４２】

コンテンツ生成時に生成したＩＣＶを格納しておき、チェック処理時に生成Ｉ

ＣＶと格納ＩＣＶの比較処理を行なう。両ＩＣＶが一致すれば改竄無しと判定し、ＩＣＶが不一致の場合は、改竄が有りと判定され、データ再生等の処理制限がなされる。

【０２４３】

メモリカード等の記憶装置には、音楽コンテンツのみならず、画像データ、ゲームプログラムデータ等、カテゴリの異なるが格納される。これら各カテゴリのコンテンツも改竄の防止を図るため、各カテゴリ毎にインテグリティ・チェック値（ＩＣＶ）を生成して格納することがコンテンツ改竄チェックのためには有効な手段となる。

【０２４４】

しかしながら、メモリに格納するコンテンツ数が増大すると、検証用のチェック値を正規のコンテンツデータに基づいて生成し、格納し管理することが困難となる。特に、昨今フラッシュメモリを使用したメモリカード等の容量の大きい媒体においては、音楽データ、画像データ、プログラムデータ等、様々なカテゴリのコンテンツデータがメモリに格納されることとなる。このような環境においては、チェック値の生成処理、格納処理、改竄チェック処理の管理は困難となる。格納データ全体に対するチェック値を生成すると、チェック対象となったデータ全体に対するチェック値生成処理を実行することが必要となる。例えばＤＥＳ－ＣＢＣモードにおいて生成されるメッセージ認証符号（ＭＡＣ）により、チェック値ＩＣＶを求める手法を行なう場合、データ全体に対するＤＥＳ－ＣＢＣの処理を実行することが必要となる。この計算量は、データ長が長くなるにつれ増大することとなり、処理効率の点で問題がある。

【０２４５】

記憶装置として使用可能なメモリカードには、多くのカテゴリの異なるコンテンツが格納される。これらのカテゴリの異なるコンテンツの改竄チェック管理をカテゴリ毎に独立したインテグリティ・チェック値（ＩＣＶ）を生成して実行する構成とすることにより、ＩＣＶのチェック時、あるいはＩＣＶの変更時、例えばデータ変更時の新たなインテグリティ・チェック値（ＩＣＶ）の生成処理が、１つのカテゴリ内のデータを対象として実行可能となり、他のカテゴリに影響を

及ぼすことがない。このようにカテゴリ毎の複数のインテグリティ・チェック値（ＩＣＶ）を格納する構成について説明する。

【０２４６】

図３７に記憶装置に格納されるデータ構成と、それぞれのインテグリティ・チェック値（ＩＣＶ）の格納構成例を示す。メモリカード等の記憶部（フラッシュメモリ）には、図３７に示されるように音楽データのディレクトリに、再生管理ファイル（ＰＢＬＩＳＴ）、暗号化コンテンツとして複数のＡＴＲＡＣＫ３データファイル（Ａ３Ｄ）が含まれ、さらに、メモリには、複数のカテゴリに属するコンテンツデータ（＃１～＃ｎ）が格納される。複数のカテゴリとは、例えば、音楽データ、画像データ、ゲームプログラム等である。さらに、同様の画像データであっても、それぞれのデータ提供元に応じて別のディレクトリとして独立のカテゴリとして管理してもよい。

【０２４７】

また、前述の有効化キーブロック（ＥＫＢ）の管理単位（エンティテイ）を１カテゴリとして設定してもよい。すなわち、ある有効化キーブロック（ＥＫＢ）によって取得されるキー暗号キー：ＫＥＫによって復号されるコンテンツキーＫｃｏｎを適用可能なコンテンツ集合を１つのカテゴリとして設定してもよい。

【０２４８】

再生管理ファイル（ＰＢＬＩＳＴ）、暗号化コンテンツとして複数のＡＴＲＡＣＫ３データファイル（Ａ３Ｄ）の各々には、改竄チェックのためのメッセージ認証符号（ＭＡＣ（Message Authentication Code））が含まれ、これらのＭＡＣ値に基づいてインテグリティ・チェック値（ＩＣＶ（ｃｏｎ））が生成される。複数のコンテンツのＭＡＣ値は、フラッシュメモリのシーケンスページにＭＡＣリストとして格納、管理され、これらのＭＡＣリストに基づいてＩＣＶ生成キーＫｉｃｖを適用して得られるインテグリティ・チェック値（ＩＣＶ（ｃｏｎ））が格納保存される。

【０２４９】

コンテンツＭＡＣ値を格納するシーケンスページフォーマットを図３８に示す。シーケンスページ領域は、一般コンテンツデータの書き込み禁止領域として設

定された領域である。図３８のシーケンスページ構成について説明する。

【０２５０】

Ｅ（ｋＳＴＲ，ｋＣＯＮ）は、メモリカードのストレージキーで暗号化したコンテンツキーである。ＩＤ（ｕｐｐｅｒ），（ｌｏｗｅｒ）は、メモリカードの識別子（ＩＤ）の格納領域である。Ｃ＿ＭＡＣ［０］は、再生管理ファイル（ＰＢＬＩＳＴ）の構成データに基づいて生成されたＭＡＣ値である。Ｃ＿ＭＡＣ［１］は、コンテンツ、例えばＡＴＲＡＣＫ３データファイル＃１のデータに基づいて生成されたＭＡＣ値、以下、コンテンツ毎にＭＡＣ値が格納される。これらのＭＡＣ値に基づいてインテグリティ・チェック値（ＩＣＶ（ｃｏｎ））が生成され、生成されたＩＣＶ（ｃｏｎ）がシリアルプロトコルを通してメモリに書き込まれる。なお、異なる鍵システムに対応するため、それぞれの鍵システムから生成されるＩＣＶをそれぞれ違うエリアに格納する構成とすることが好ましい。

【０２５１】

また、カテゴリ毎に改竄チェックのために生成される各カテゴリ毎のインテグリティ・チェック値（ＩＣＶ）は、メモリカードの記憶部（フラッシュメモリ）のプールページに記録される。プールページもまた、一般データの書き込みの禁止された領域として設定されている。

【０２５２】

各カテゴリ毎のインテグリティ・チェック値（ＩＣＶ）を格納するプールページフォーマットを図３９に示す。＃０＿ｒｅｖｉｓｉｏｎは、カテゴリ＃０の更新データが設定され、更新された場合はインクリメントされる。＃０＿ｖｅｒｓｉｏｎは、カテゴリ＃０のバージョン、＃０＿Ｅ（ＫＥＫ，Ｋｉｃｖ）は、カテゴリ＃０のキー暗号化キー（ＫＥＫ）で暗号化したＩＣＶ生成キー（Ｋｉｃｖ）であり、ＩＣＶ０は、カテゴリ＃０のインテグリティ・チェック値（ＩＣＶ）値である。以下、同様のデータが各カテゴリ毎にＥＫＢ＃１５まで格納可能となっている。

【０２５３】

ＩＣＶのチェックは、パワーオン時、またはメモリカード等の記憶装置が再生装置にセットされたことを条件として開始される。図４０にＩＣＶチェックを含

む処理フローを示す。

【０２５４】

まず、再生装置がパワーオン、または新たなメモリカード等が装着されたことを検知すると、ステップＳ４００１において、再生装置と記憶装置間の相互認証が可能か否かが判定され、可能である場合は、ステップＳ４００２において記憶装置と再生装置間での相互認証処理（図２８参照）が実行される。また、ステップＳ４００１において、再生装置と記憶装置間の相互認証が可能でないと判定された場合は、ステップＳ４００３において、前述した仮想メモリカードと再生装置間の相互認証処理が実行される。

【０２５５】

ステップＳ４００４で相互認証が成立したか否かが判定され、不成立の場合は、以下の処理は実行されないで終了する。相互認証が成立の場合は、ステップＳ４００５においてＩＣＶの計算が実行される。ＩＣＶは、前述したように各ファイルのＭＡＣ値に基づいて算出される。

【０２５６】

次にステップＳ４００６において、計算によって算出された生成ＩＣＶと、予め格納してある格納ＩＣＶとの比較が実行される。両ＩＣＶが一致した場合は、データ改竄がないと判定され、ステップＳ４００７において、データ再生等の様々な処理が実行される。一方、ＩＣＶが不一致であった場合は、データ改竄があると判定され、データの再生等を行なわず処理を終了する。このような処理を実行することによりデータ改竄の防止、改竄されたデータの再生が排除される。

【０２５７】

このように、カテゴリの異なるコンテンツについて、カテゴリ毎に独立したインテグリティ・チェック値（ＩＣＶ）を生成して管理する構成とすることにより、ＩＣＶのチェック時、あるいはＩＣＶの変更時、例えばデータ変更時の新たなインテグリティ・チェック値（ＩＣＶ）の生成処理が、１つのカテゴリ内のデータを対象として実行可能となり、他のカテゴリに影響を及ぼすことがない。

【０２５８】

以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしな

がら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参酌すべきである。

【０２５９】

【発明の効果】

以上、説明したように、本発明の情報処理装置、および方法によれば、記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行する構成において、検証値をコンテンツのカテゴリ毎に独立した検証値として生成して格納する構成としたので、コンテンツデータの検証処理の効率化、さらに検証後の記録デバイスに対するダウンロード処理、あるいは検証後の再生処理等を効率的に実行することが可能となる。

【０２６０】

さらに、本発明の情報処理装置、および方法によれば、カテゴリは、コンテンツの種類、あるいは、コンテンツの暗号処理鍵として設定されるコンテンツキーＫｃｏｎを暗号化して提供する有効化キーブロック（ＥＫＢ）の管理エンティテイに基づいて設定された構成としたので、例えば有効化キーブロック（ＥＫＢ）の管理エンティテイ別に、コンテンツデータの検証処理を独立して実行可能となり、処理の効率化が実現される。

【図面の簡単な説明】

【図１】

本発明の情報処理システムの使用概念を説明する図である。

【図２】

本発明の情報処理システムのシステム構成例およびデータ経路例を示す図である。

【図３】

本発明の情報処理システムにおける各種キー、データの暗号化処理について説明するツリー構成図である。

【図４】

本発明の情報処理システムにおける各種キー、データの配布に使用される有効化キーブロック（ＥＫＢ）の例を示す図である。

【図５】

本発明の情報処理システムにおけるコンテンツキーの有効化キーブロック（ＥＫＢ）を使用した配布例と復号処理例を示す図である。

【図６】

本発明の情報処理システムにおける有効化キーブロック（ＥＫＢ）のフォーマット例を示す図である。

【図７】

本発明の情報処理システムにおける有効化キーブロック（ＥＫＢ）のタグの構成を説明する図である。

【図８】

本発明の情報処理システムにおける有効化キーブロック（ＥＫＢ）と、コンテンツキー、コンテンツを併せて配信するデータ構成例を示す図である。

【図９】

本発明の情報処理システムにおける有効化キーブロック（ＥＫＢ）と、コンテンツキー、コンテンツを併せて配信した場合のデバイスでの処理例を示す図である。

【図１０】

本発明の情報処理システムにおける有効化キーブロック（ＥＫＢ）とコンテンツを記録媒体に格納した場合の対応について説明する図である。

【図１１】

本発明の情報処理システムにおける階層ツリー構造のカテゴリ分類の例を説明する図である。

【図１２】

本発明の情報処理システムにおける簡略化有効化キーブロック（ＥＫＢ）の生成過程を説明する図である。

【図１３】

本発明の情報処理システムにおける有効化キーブロック（ＥＫＢ）の生成過程を説明する図である。

【図１４】

本発明の情報処理システムにおける簡略化有効化キーブロック（ＥＫＢ）を説明する図である。

【図１５】

本発明の情報処理システムにおける再生装置と記憶装置の構成を示すブロック図である。

【図１６】

本発明の情報処理システムにおける記憶装置内の記憶ユニットに記憶されているデータを説明する図である。

【図１７】

本発明の情報処理システムにおける記憶装置のフラッシュメモリに記憶されるデータを説明するための図である。

【図１８】

本発明の情報処理システムにおける再生管理ファイルのデータ構成を概略的に示す図である。

【図１９】

本発明の情報処理システムにおけるデータファイルのデータ構成を概略的に示す図である。

【図２０】

本発明の情報処理システムにおけるデータファイルのデータ構成をより詳細に示す図である。

【図２１】

本発明の情報処理システムにおけるデータファイルの属性ヘッダの一部を示す図である。

【図２２】

本発明の情報処理システムにおけるデータファイルの属性ヘッダの一部を示す図である。

【図２３】

本発明の情報処理システムにおけるモードの種類と、各モードにおける録音時間等を示す図である。

【図２４】

本発明の情報処理システムにおけるコピー制御情報を説明するための図である。

【図２５】

本発明の情報処理システムにおけるデータファイルの属性ヘッダの一部を示す図である。

【図２６】

本発明の情報処理システムにおけるデータファイルの各データブロックのヘッダを示す略線図である。

【図２７】

本発明の情報処理システムにおけるデータ記録処理フローを示す図である。

【図２８】

本発明の情報処理システムにおいて適用可能な相互認証処理を示す図である。

【図２９】

本発明の情報処理システムにおけるデータ再生処理フローを示す図である。

【図３０】

本発明の情報処理システムにおける配信鍵許可情報ファイルのフオーマットを示す図である。

【図３１】

本発明の情報処理システムにおけるデータ格納態様を示す図である。

【図３２】

本発明の情報処理システムにおけるキー有効化ブロック（ＥＫＢ）を使用したデータ復号処理フローを示す図である。

【図３３】

本発明の情報処理システムにおける有効化キーブロック（ＥＫＢ）と、認証キーを併せて配信するデータ構成と、デバイスでの処理例を示す図（その１）であ

る。

【図３４】

本発明の情報処理システムにおける有効化キーブロック（ＥＫＢ）と、認証キーを併せて配信するデータ構成と、デバイスでの処理例を示す図（その２）である。

【図３５】

本発明の情報処理システムにおける仮想メモリカードを適用したて認証処理シーケンスを示す図である。

【図３６】

本発明の情報処理システムにおいて適用可能なインテグリティ・チェック値（ＩＣＶ）の生成に使用するＭＡＣ値生成例を示す図である。

【図３７】

本発明の情報処理システムにおけるインテグリティ・チェック値（ＩＣＶ）の格納態様を説明する図である。

【図３８】

本発明の情報処理システムにおけるＭＡＣ値を格納するシーケンスページフォーマットを示す図である。

【図３９】

本発明の情報処理システムにおけるＩＣＶを格納するプールページフォーマットを示す図である。

【図４０】

本発明の情報処理システムにおけるＩＣＶチェック処理フローを示す図である。

【符号の説明】

１０　コンテンツ配信手段

１１　インターネット

１２　衛星放送

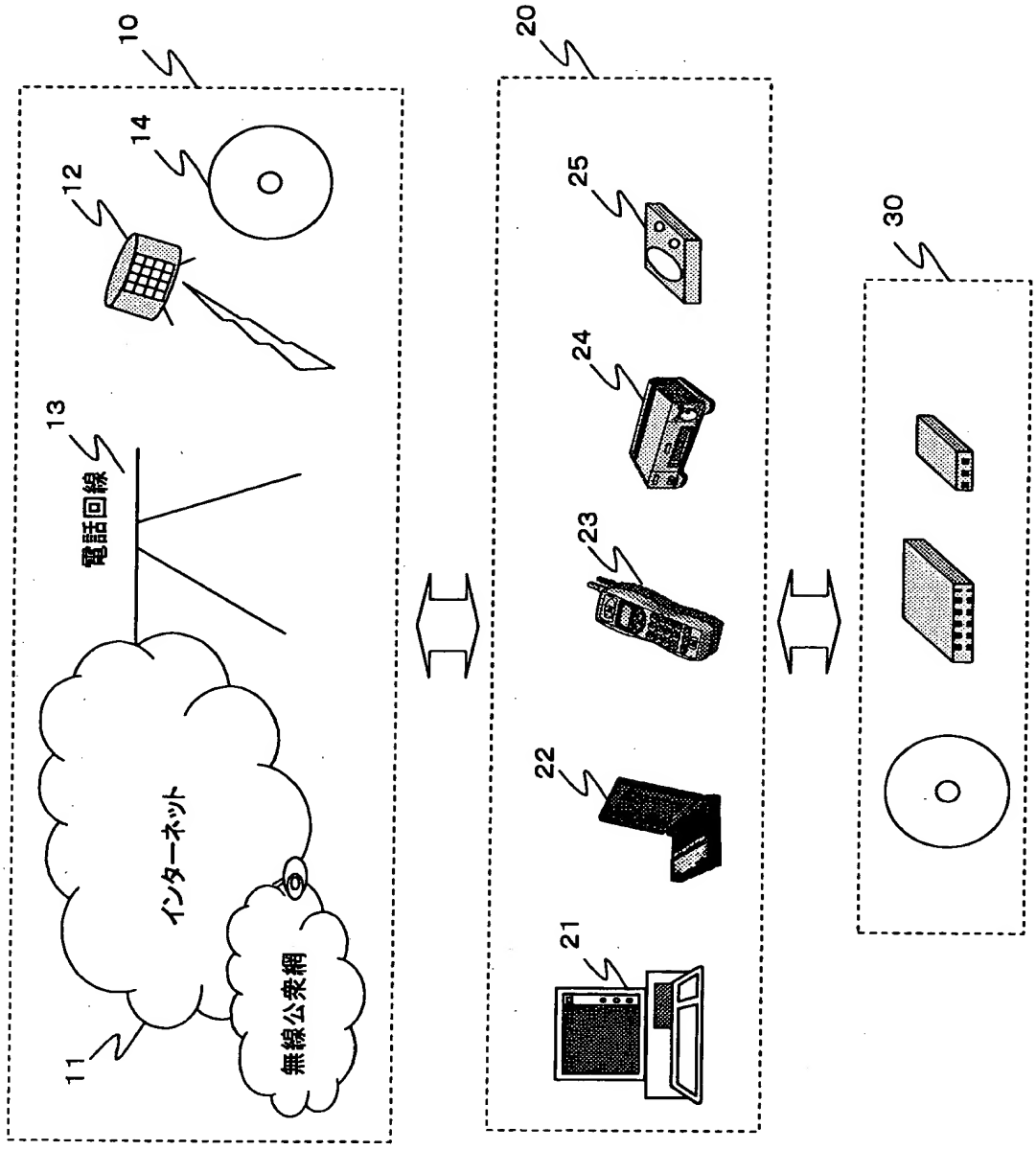
１３　電話回線

１４　メディア

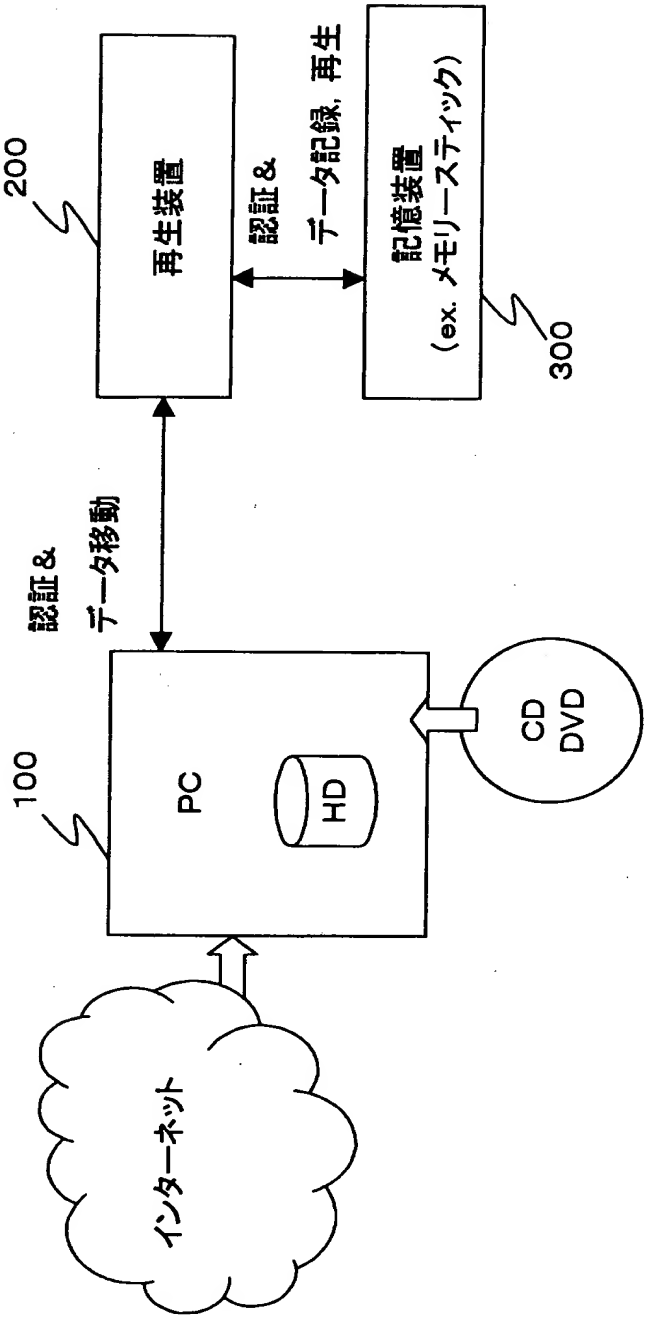
２０　データ処理手段
２１　パーソナルコンピュータ（ＰＣ）
２２　ポータブルデバイス（ＰＤ）
２３　携帯電話、ＰＤＡ
２４　記録再生器、ゲーム端末
２５　再生装置
３０　記憶手段
１００　パーソナルコンピュータ（ＰＣ）
２００　再生装置
３００　記憶装置
６０１　バージョン
６０２　デプス
６０３　データポインタ
６０４　タグポインタ
６０５　署名ポインタ
６０６　データ部
６０７　タグ部
６０８　署名
３３，４３　制御モジュール
５０，６０　乱数発生ユニット
５１，６１　記憶ユニット
５２，６２　鍵生成／演算ユニット
５３，６３　相互認証ユニット
５４，７４　暗号化／復号ユニット
５５，６５　制御ユニット
３４　フラッシュメモリ
４４　編集モジュール
４５　圧縮／伸長モジュール
４６　スピーカ

【書類名】　　図面

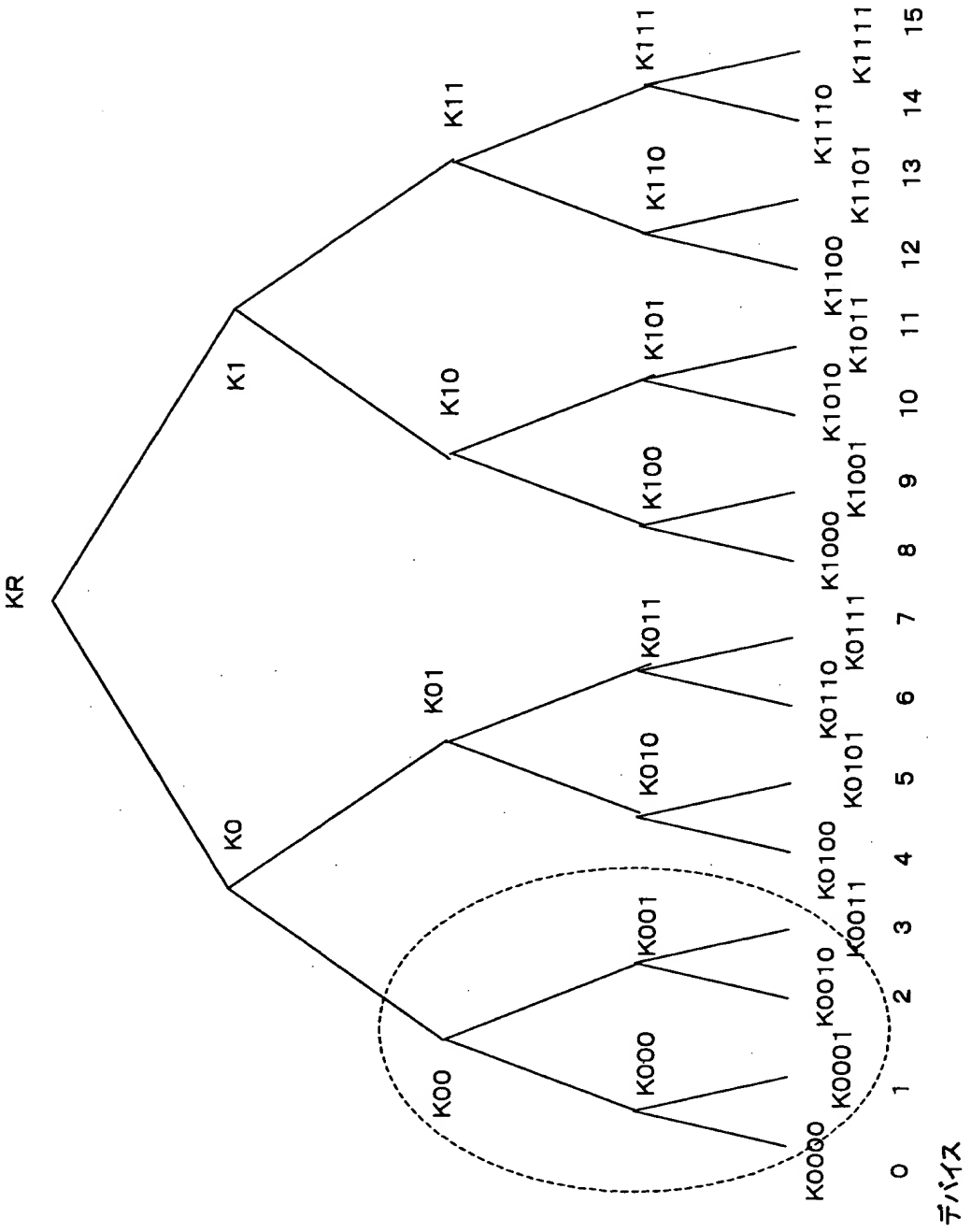
【図１】



【図２】


200
再生装置
認証＆
データ記録，再生
記憶装置
(ex. メモリースティック)
300
認証＆
データ移動
100
PC
HD
CD
DVD
インターネット

【図３】


KR
K0
K1
K00
K01
K10
K11
K000
K001
K010
K011
K100
K101
K110
K111
K0000
K0001
K0010
K0011
K0100
K0101
K0110
K0111
K1000
K1001
K1010
K1011
K1100
K1101
K1110
K1111
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
デバイス

【図４】

（A） 有効化キーブロック(EKB:Enabling Key Block)　例1

デバイス0, 1, 2にバージョン:tのノードキーを送付

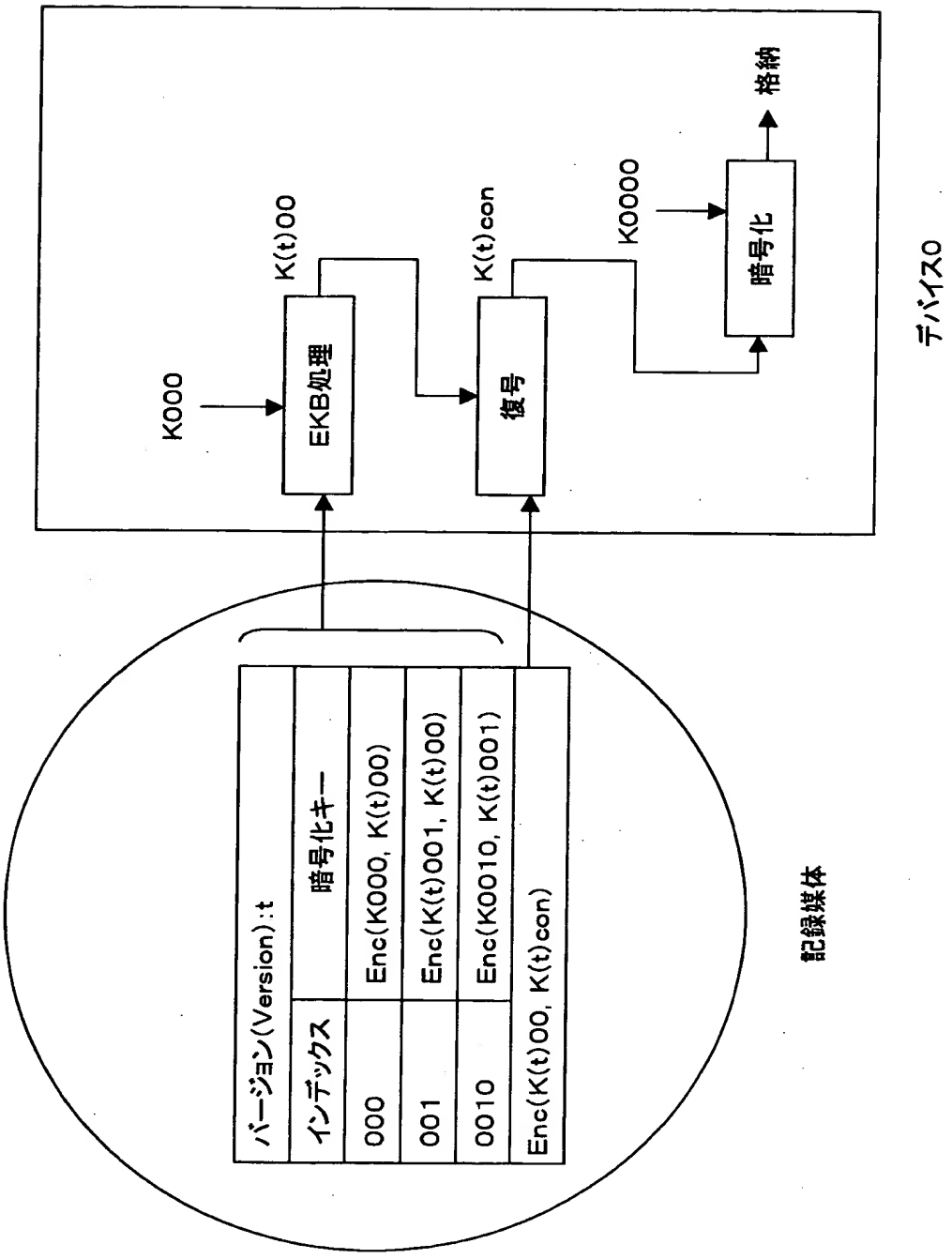
バージョン(Version):t	
インデックス	暗号化キー
0	Enc(K(t)0, K(t)R)
00	Enc(K(t)00, K(t)0)
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

（B） 有効化キーブロック(EKB:Enabling Key Block)　例2

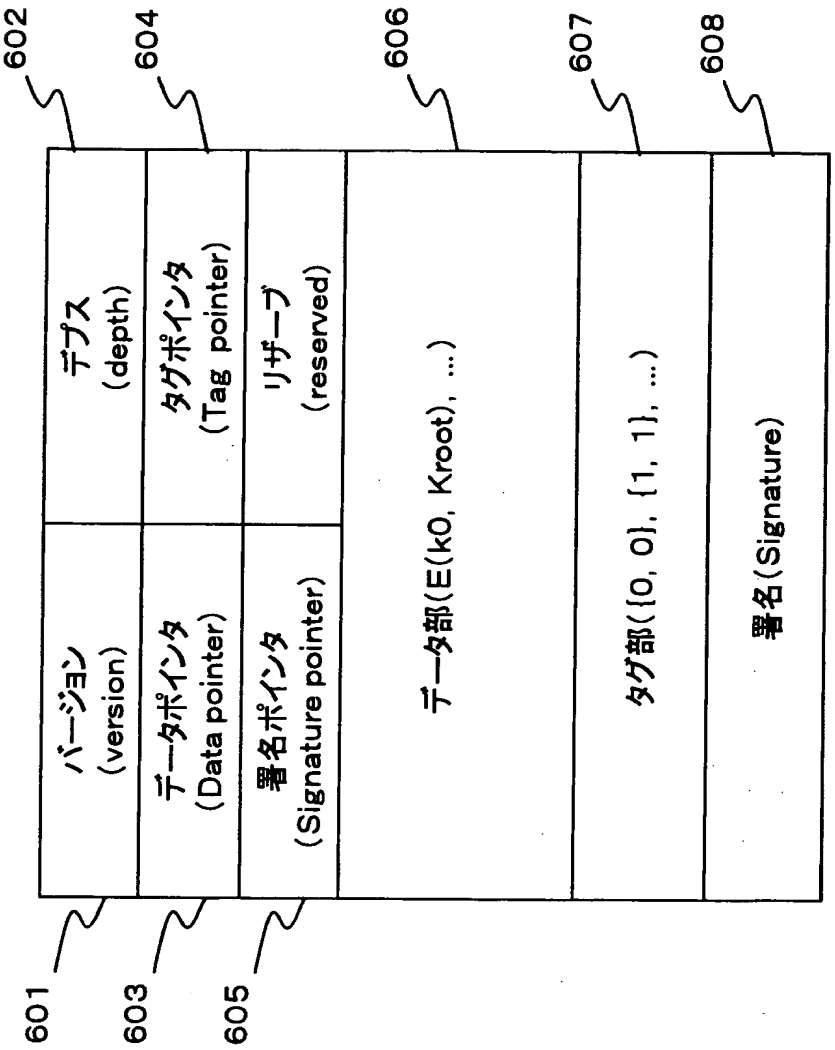
デバイス0, 1, 2にバージョン:tのノードキーを送付

バージョン(Version):t	
インデックス	暗号化キー
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

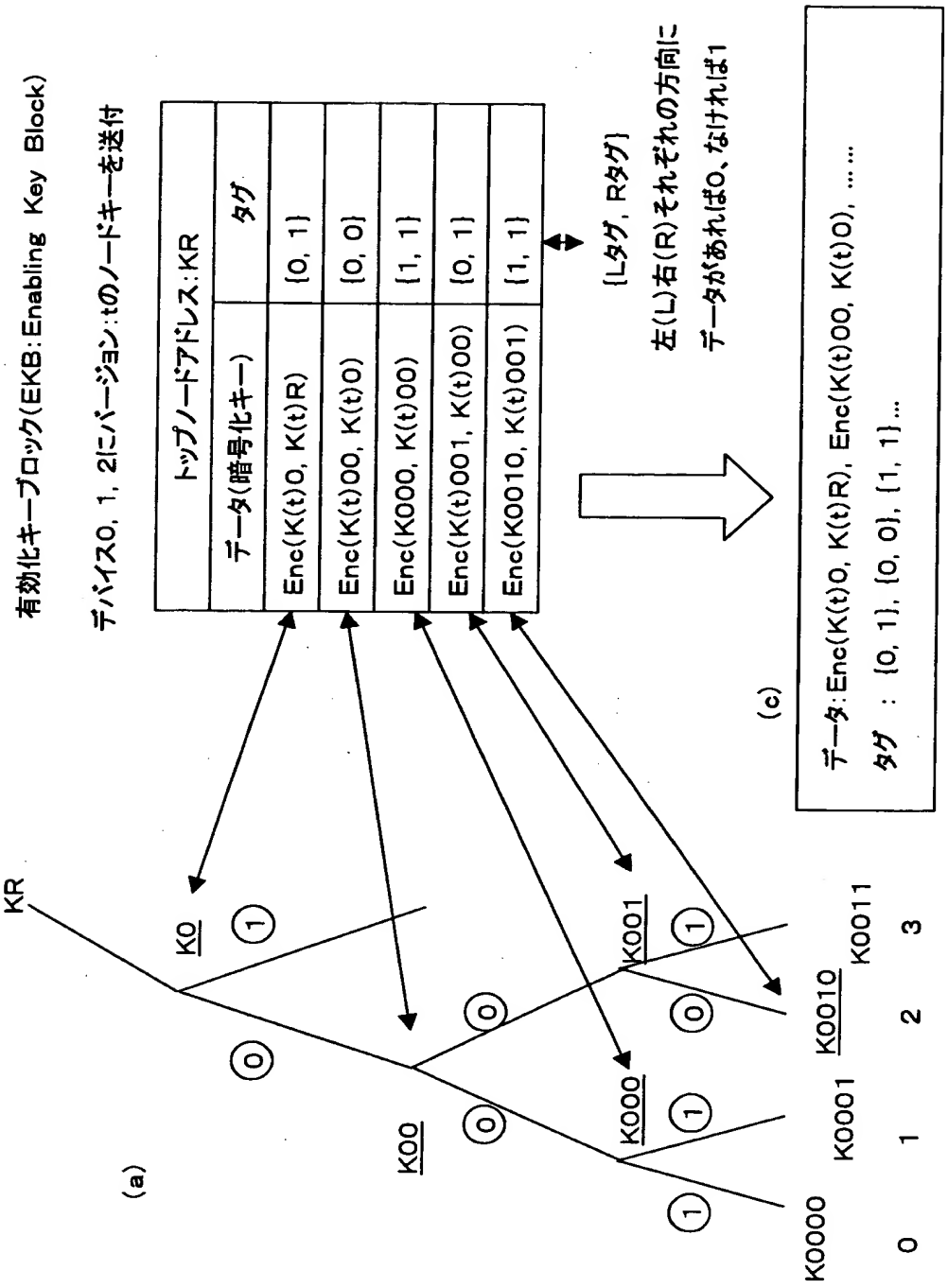
【図５】


K000
EKB処理
K(t)00
復号
K(t)con
K0000
暗号化
格納
デバイス0
記録媒体
バージョン(Version):t
インデックス
暗号化キー
000
Enc(K000, K(t)00)
001
Enc(K(t)001, K(t)00)
0010
Enc(K0010, K(t)001)
Enc(K(t)00, K(t)con)

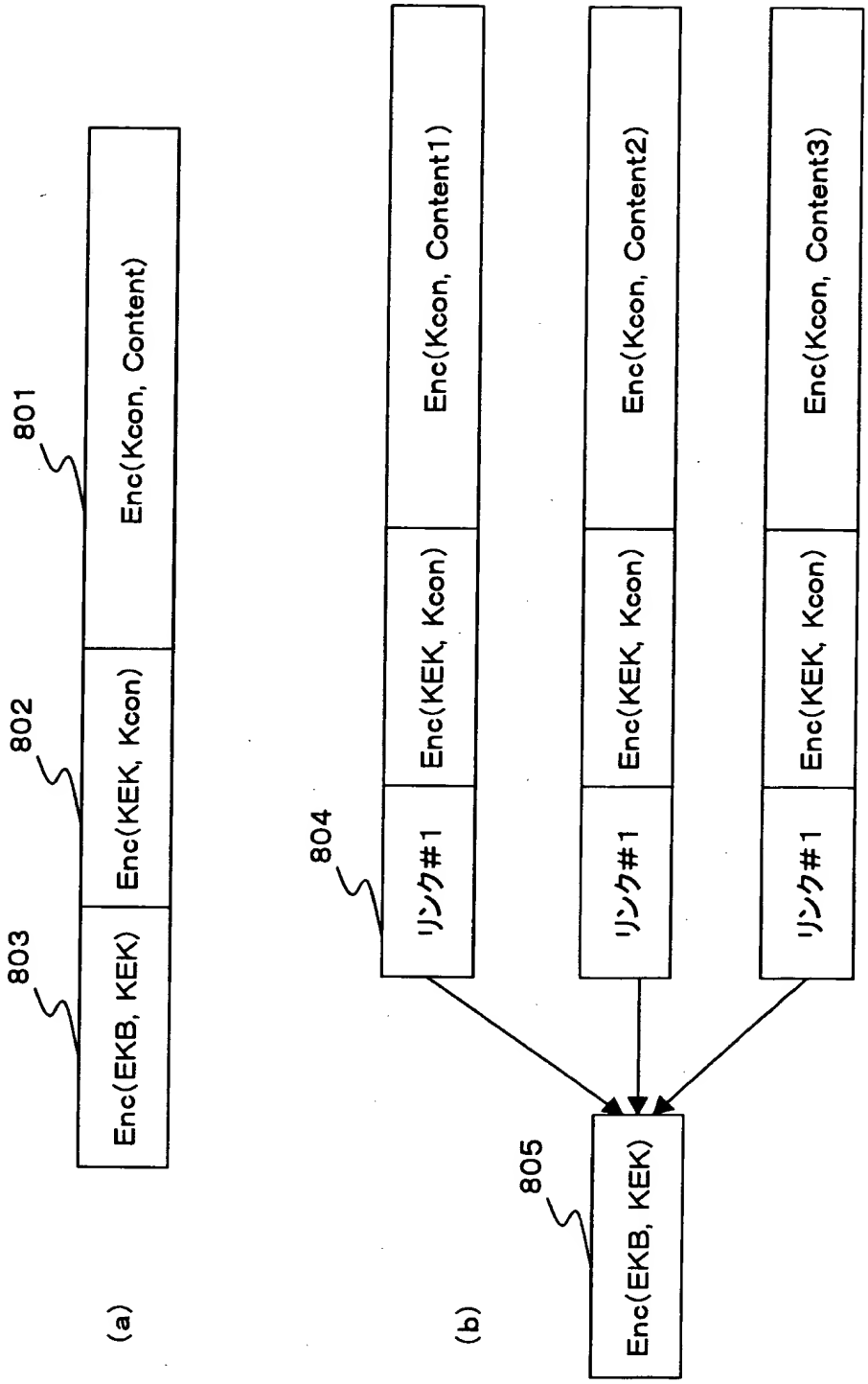
【図６】


602
604
606
607
608
デプス (depth)
タグポインタ (Tag pointer)
リザーブ (reserved)
データ部(E(k0, Kroot), ...)
タグ部({0, 0}, {1, 1}, ...)
署名(Signature)
バージョン (version)
データポインタ (Data pointer)
署名ポインタ (Signature pointer)
601
603
605

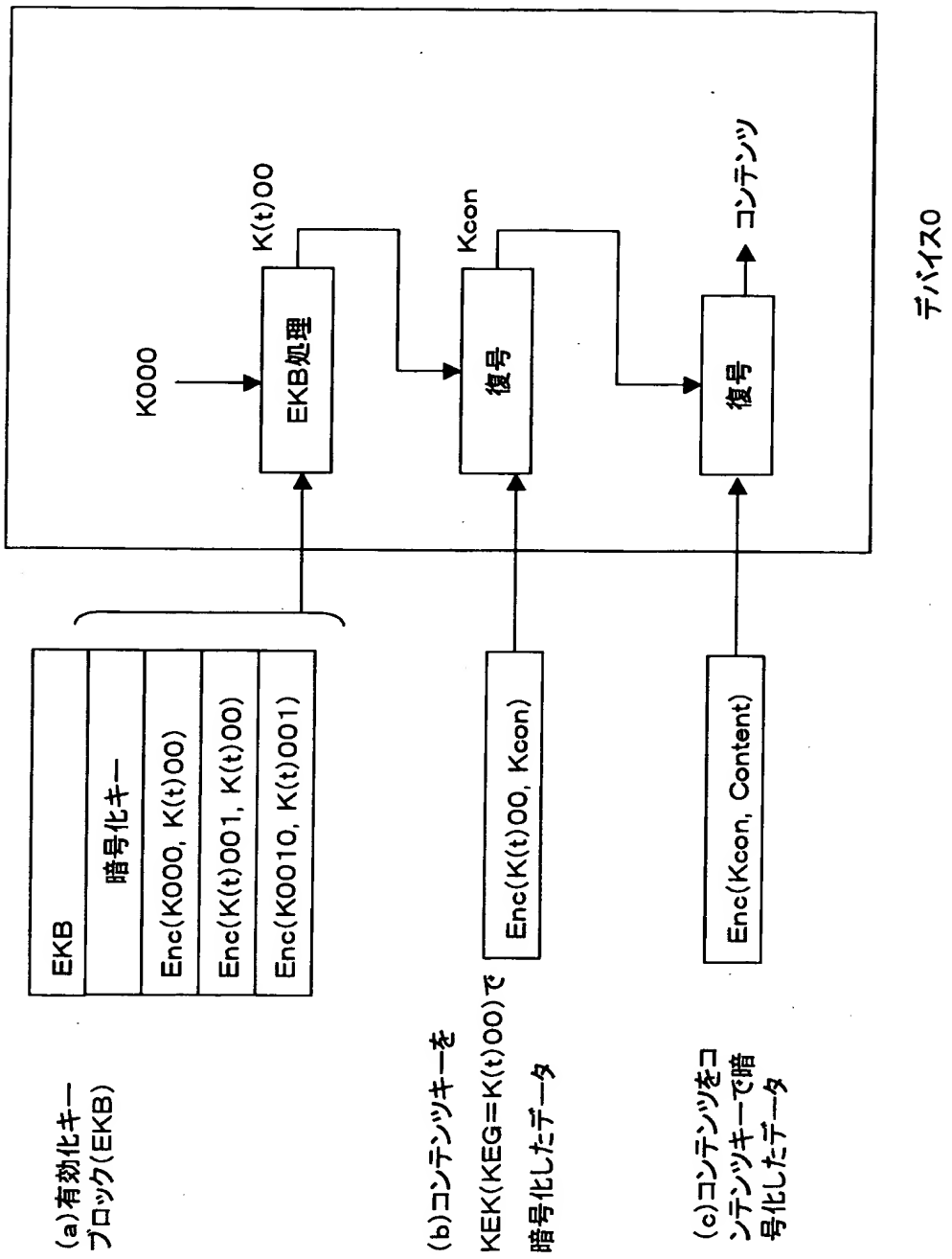
【図７】


(a)
KR
K0
K00
K000
K0000
K0001
K001
K0010
K0011
0
1
2
3
(b)
有効化キーブロック(EKB:Enabling Key Block)
デバイス0, 1, 2にバージョン:tのノードキーを送付
トップノードアドレス:KR
データ(暗号化キー)
タグ
Enc(K(t)0, K(t)R)
{0, 1}
Enc(K(t)00, K(t)0)
{0, 0}
Enc(K000, K(t)00)
{1, 1}
Enc(K(t)001, K(t)00)
{0, 1}
Enc(K0010, K(t)001)
{1, 1}
{Lタグ, Rタグ}
左(L)右(R)それぞれの方向に
データがあれば0、なければ1
(c)
データ:Enc(K(t)0, K(t)R), Enc(K(t)00, K(t)0), ……
タグ : {0, 1}, {0, 0}, {1, 1}…

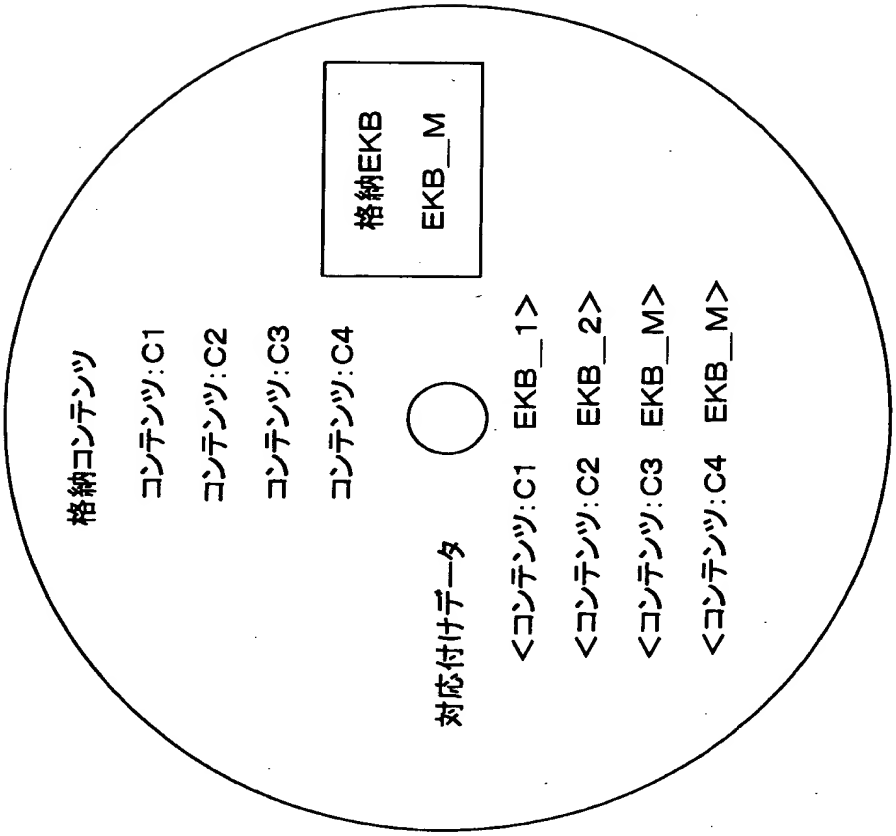
【図８】


801
Enc(Kcon, Content)
802
Enc(KEK, Kcon)
803
Enc(EKB, KEK)
(a)
804
リンク#1
Enc(KEK, Kcon)
Enc(Kcon, Content1)
リンク#1
Enc(KEK, Kcon)
Enc(Kcon, Content2)
リンク#1
Enc(KEK, Kcon)
Enc(Kcon, Content3)
805
Enc(EKB, KEK)
(b)

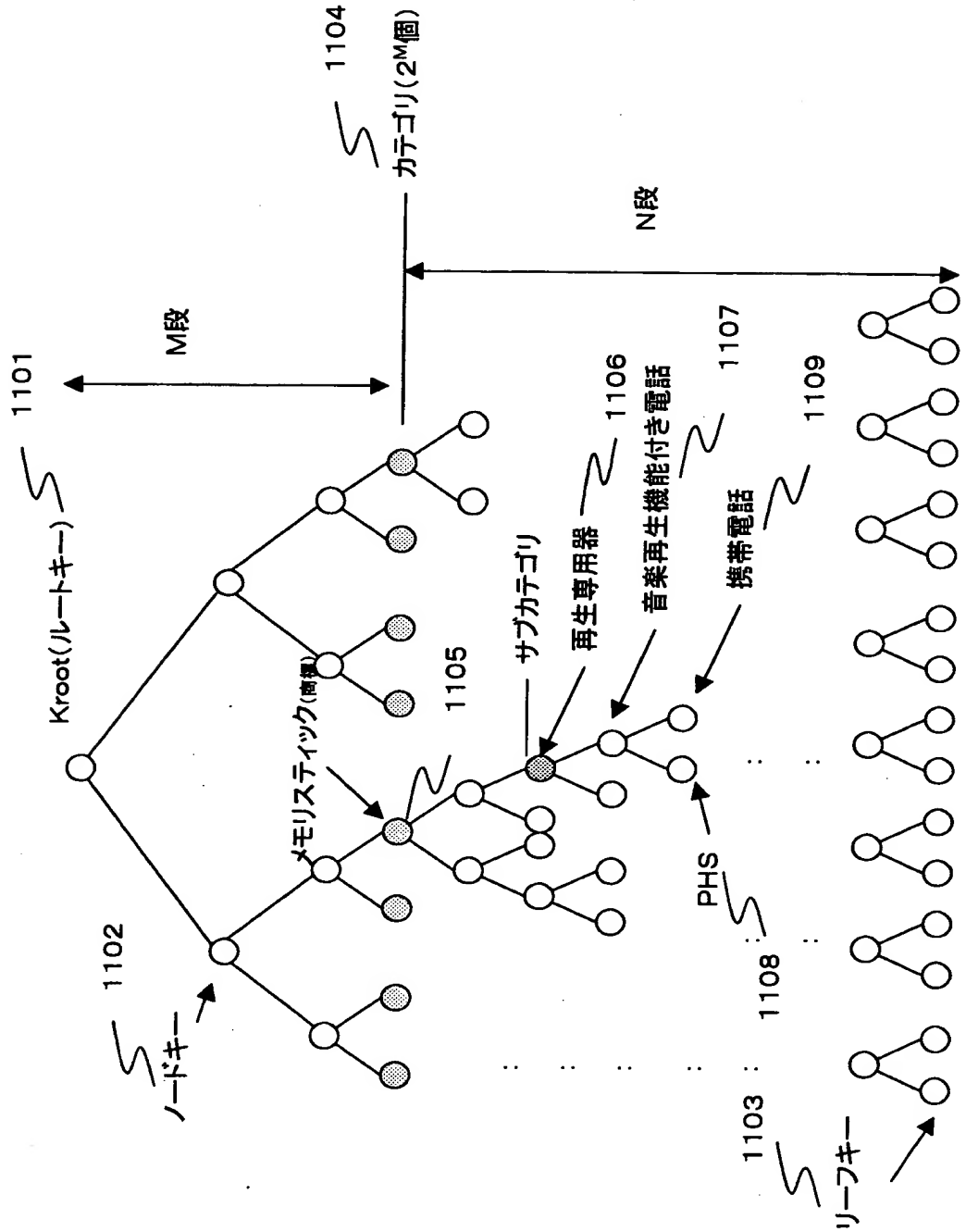
【図９】


K000
EKB処理
K(t)00
復号
Kcon
復号
コンテンツ
デバイス0
EKB
暗号化キー
Enc(K000, K(t)00)
Enc(K(t)001, K(t)00)
Enc(K0010, K(t)001)
Enc(K(t)00, Kcon)
Enc(Kcon, Content)
(a)有効化キーブロック(EKB)
(b)コンテンツキーをKEK(KEG=K(t)00)で暗号化したデータ
(c)コンテンツをコンテンツキーで暗号化したデータ

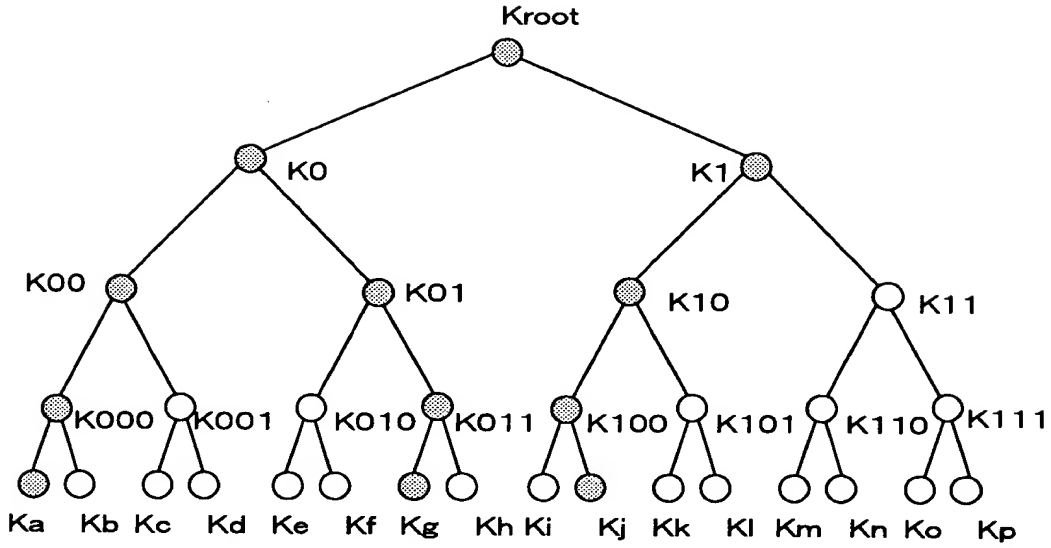
【図１０】


格納コンテンツ
コンテンツ:C1
コンテンツ:C2
コンテンツ:C3
コンテンツ:C4
格納EKB
EKB_M
対応付けデータ
<コンテンツ:C1 EKB_1>
<コンテンツ:C2 EKB_2>
<コンテンツ:C3 EKB_M>
<コンテンツ:C4 EKB_M>
記録媒体

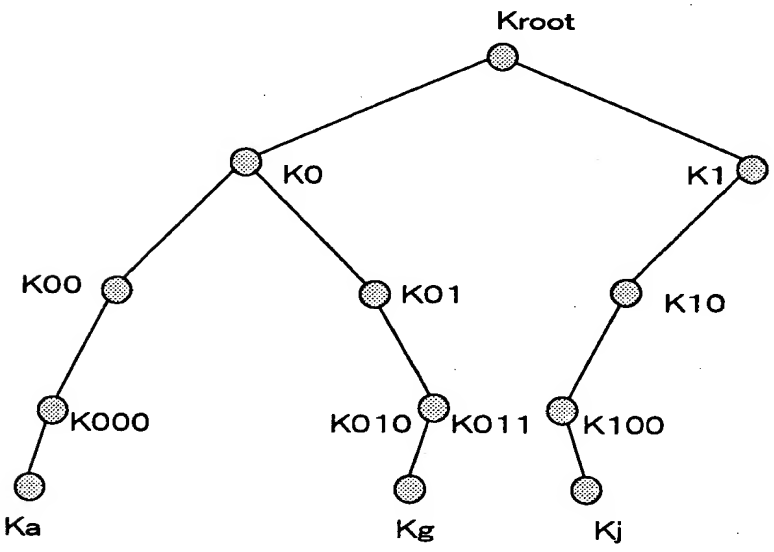
【図１１】


1104
カテゴリ(2M個)
N段
M段
1101
Kroot(ルートキー)
1107
1109
1106
音楽再生機能付き電話
携帯電話
再生専用器
サブカテゴリ
1105
メモリスティック(商標)
PHS
1102
ノードキー
1108
1103
リーフキー

【図１２】


Kroot
K0
K1
K00
K01
K10
K11
K000
K001
K010
K011
K100
K101
K110
K111
Ka
Kb
Kc
Kd
Ke
Kf
Kg
Kh
Ki
Kj
Kk
Kl
Km
Kn
Ko
Kp

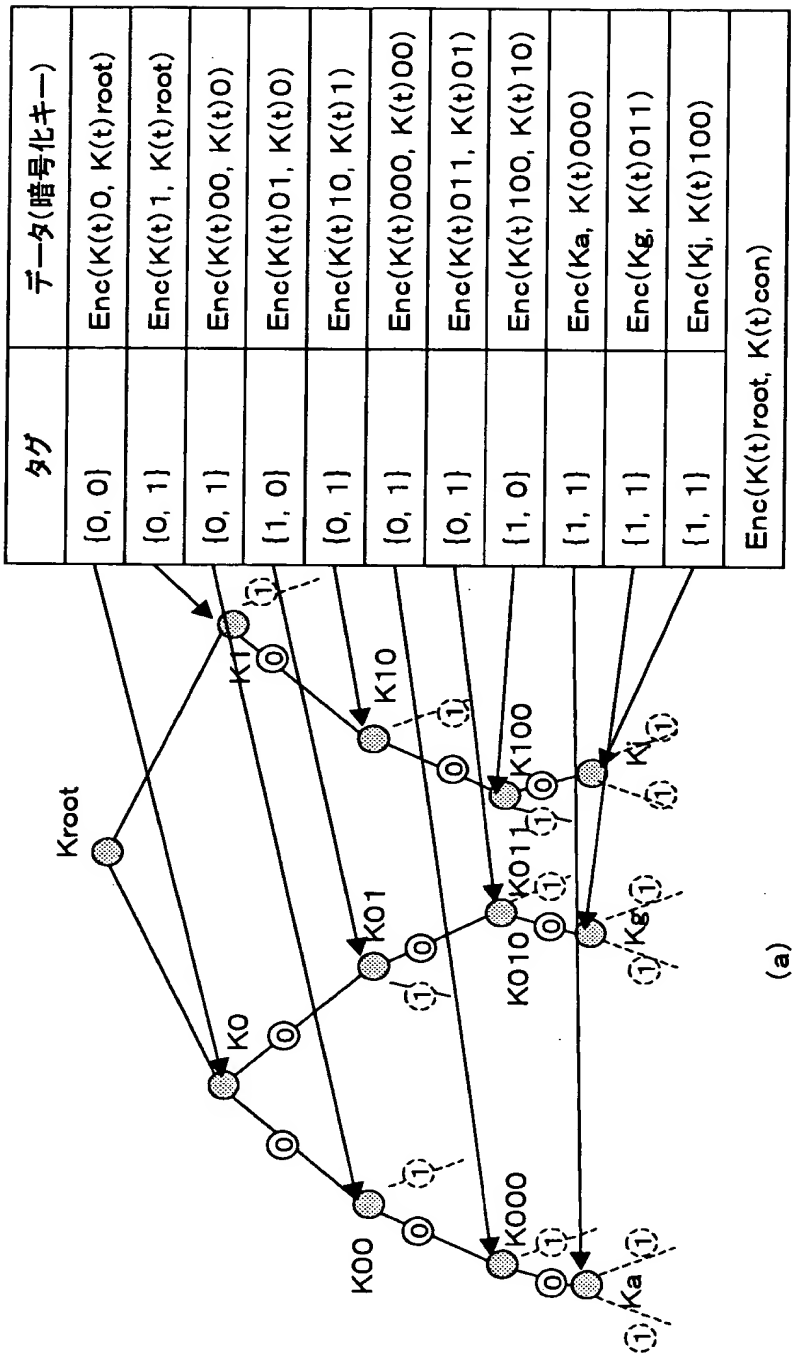
(a)


Kroot
K0
K1
K00
K01
K10
K000
K010
K011
K100
Ka
Kg
Kj

(b)

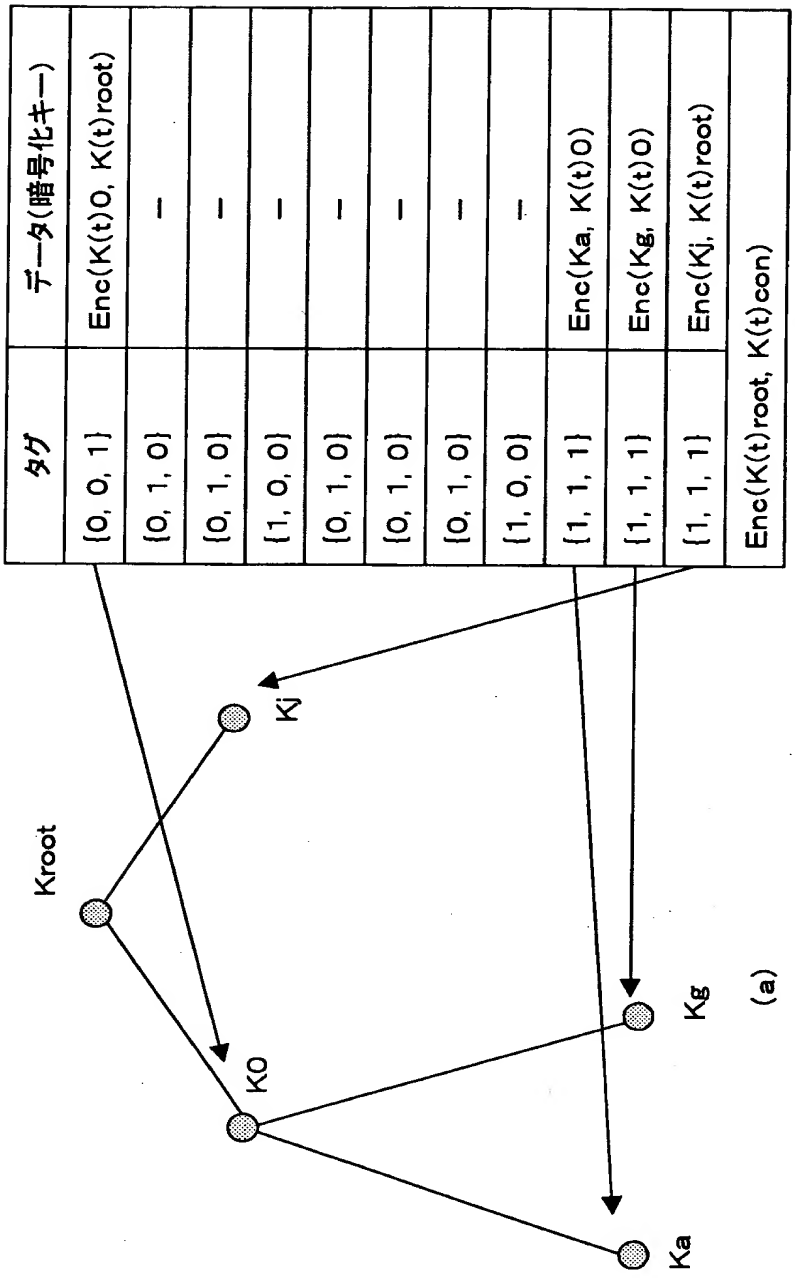
【図１３】

有効化キーブロック(EKB:Enabling Key Block)を用いた
デバイスKa, Kg, Kjへのバージョンｔのコンテンツキー送付処理

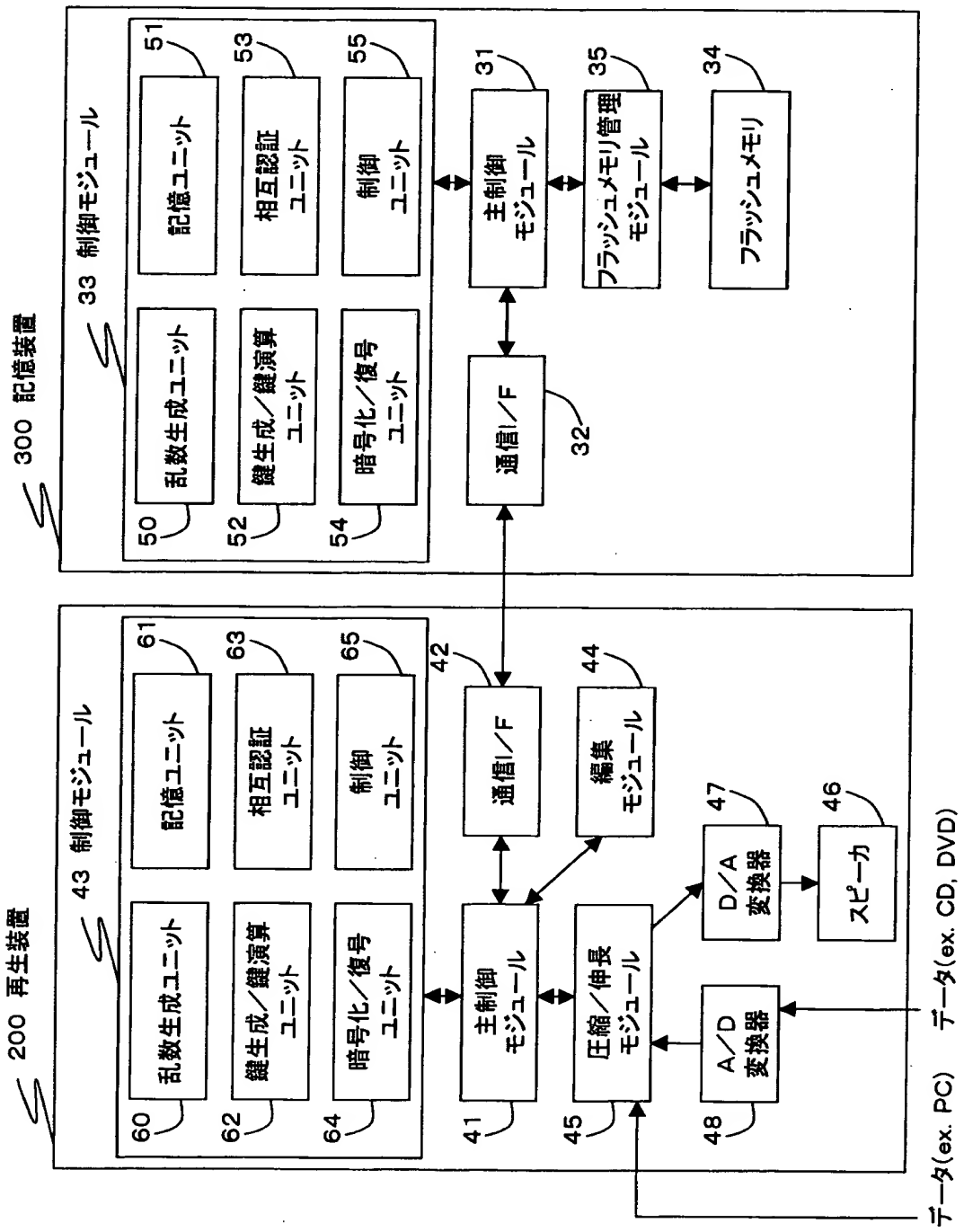


【図１４】

簡略化した有効化キーブロック(EKB:Enabling Key Block)を用いた
デバイスKa, Kg, Kjへのバージョンｔのコンテンツキー送付処理



【図１５】

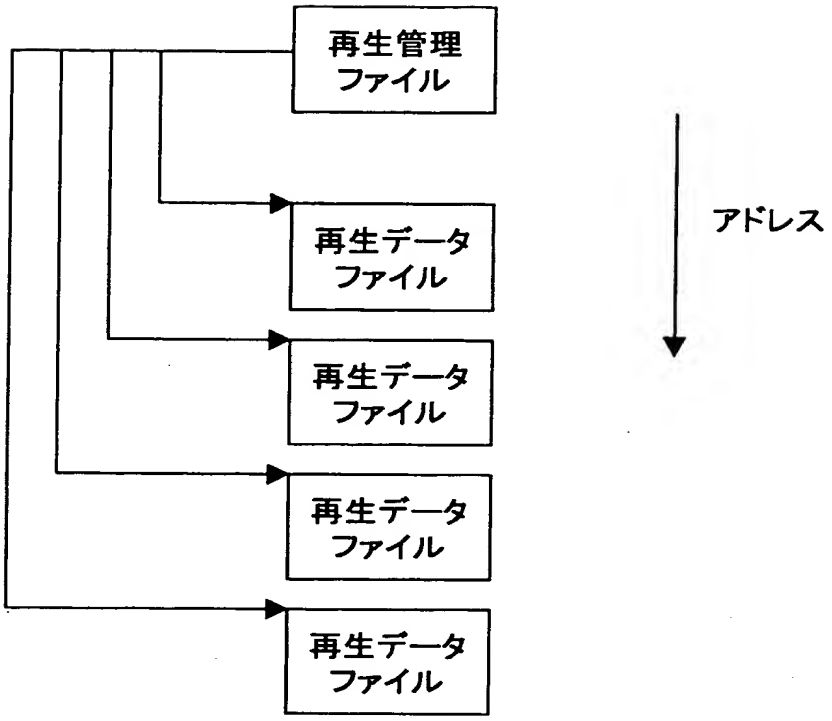

300 記憶装置
33 制御モジュール
51
記憶ユニット
53
相互認証ユニット
55
制御ユニット
50
乱数生成ユニット
52
鍵生成／鍵演算ユニット
54
暗号化／復号ユニット
31
主制御モジュール
35
フラッシュメモリ管理モジュール
34
フラッシュメモリ
32
通信I／F
200 再生装置
43 制御モジュール
61
記憶ユニット
63
相互認証ユニット
65
制御ユニット
60
乱数生成ユニット
62
鍵生成／鍵演算ユニット
64
暗号化／復号ユニット
42
通信I／F
44
編集モジュール
41
主制御モジュール
45
圧縮／伸長モジュール
47
D／A変換器
46
スピーカ
48
A／D変換器
データ(ex. PC)
データ(ex. CD, DVD)

【図１６】

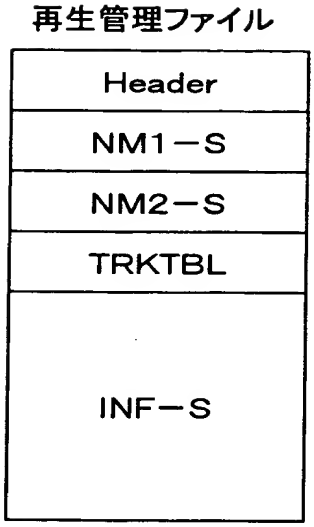
記憶装置の記憶ユニットに格納されるデータ

認証鍵データ	IK0
	IK1
	IK2
	IK3
	:
	:
	IK30
	IK31
装置識別データ	ID0
記憶用鍵データ	Kstr

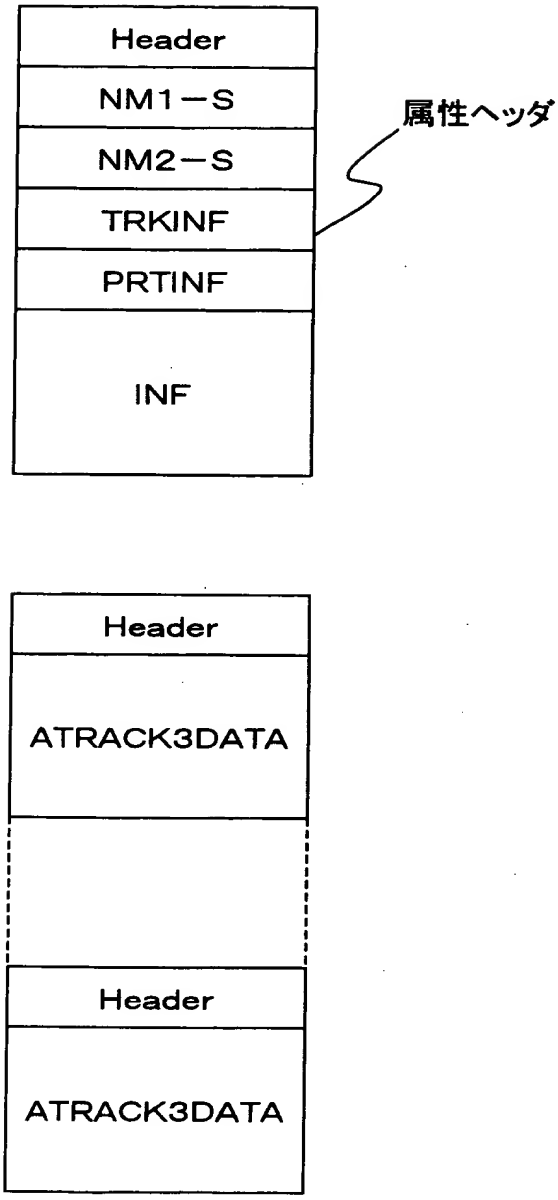
【図１７】



【図１８】



【図１９】



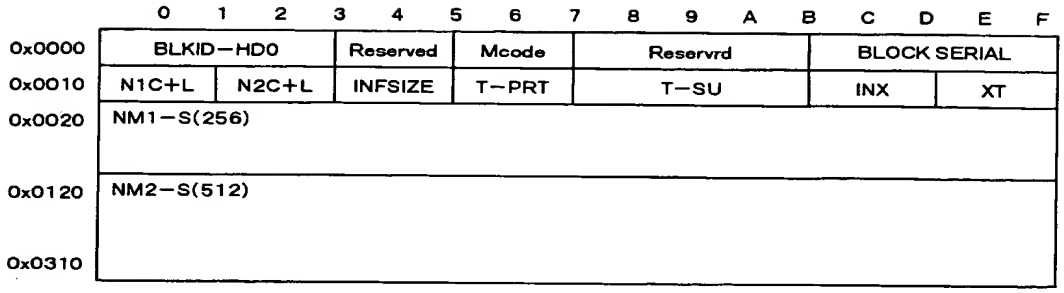
出証特２００１－３０３８６４２

【図２０】

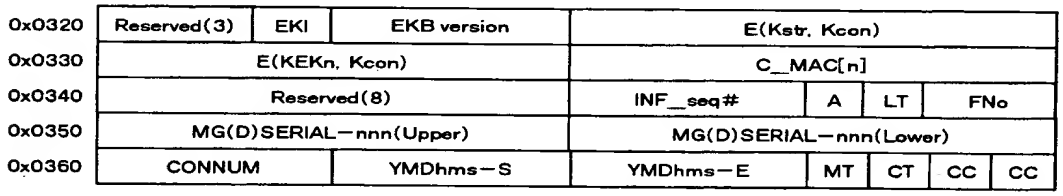
ATRAC3データファイル

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x0000	BLKID－HD0			Reserved		Mcode		Reservrd					BLOCK SERIAL			
0x0010	N1C+L		N2C+L	INFSIZE		T－PRT		T－SU					INX		XT	
0x0020	NM1－S(256)															
0x0120	NM2－S(512)															
0x0310																
0x0320	Reserved(3)		EKI	EKB version				E(Kstr, Kcon)								
0x0330	E(KEKn, Kcon)							C_MAC[n]								
0x0340	Reserved(8)							INF_seq#				A	LT	FNo		
0x0350	MG(D)SERIAL－nnn(Upper)							MG(D)SERIAL－nnn(Lower)								
0x0360	CONNUM			YMDhms－S				YMDhms－E				MT	CT	CC	CC	
0x0370	PRTSIZE			PRTKEY								Reserved(8)				
0x0380				CONNUM0				PRTSIZE(0x0388)				PRTKEY				
0x0390				Reserved(8)								CONNUM0				
	INF(0x0400)															
0x3FFF	BLKID－HD0			Reserved		Mcode		Reservrd				BLOCK SERIAL				
0x4000	BLKID－A3D			Reserved		Mcode		CONNUM0				BLOCK SERIAL				
0x4010	BLOCKSEED							INITIALIZATION VECTOR								
0x4020	SU－000(Nbyte＝384byte)															
0x41A0	SU－001(Nbyte)															
0x4320	SU－002(Nbyte)															
0x04A0	SU－041(Nbyte)															
0x7DA0	Reserved(Nbyte＝208byte)															
0x7F20	BLK SEED															
0x7FF0	BLKID－A3D			Reserved		Mcode		CONNUM0				BLOCK SERIAL				

【図２１】


0 1 2 3 4 5 6 7 8 9 A B C D E F
0x0000
BLKID－HD0
Reserved
Mcode
Reservrd
BLOCK SERIAL
0x0010
N1C+L
N2C+L
INFSIZE
T－PRT
T－SU
INX
XT
0x0020
NM1－S(256)
0x0120
NM2－S(512)
0x0310

【図２２】


0x0320
Reserved(3)
EKI
EKB version
E(Kstr, Kcon)
0x0330
E(KEKn, Kcon)
C_MAC[n]
0x0340
Reserved(8)
INF_seq#
A
LT
FNo
0x0350
MG(D)SERIAL－nnn(Upper)
MG(D)SERIAL－nnn(Lower)
0x0360
CONNUM
YMDhms－S
YMDhms－E
MT
CT
CC
CC

【図２３】

bit7：ATRACK3のモード　0：Dual　1：Joint

bit6，5，4：3bitのNはモードの値

N	モード	時間	転送レート	SU	バイト
7	HQ	47min	176kbps	31SU	512
6		58min	146kbps	38SU	424
5	EX	64min	132kbps	42SU	384
4	SP	81min	105kbps	53SU	304
3		90min	94kbps	59SU	272
2	LP	128min	66kbps	84SU	192
1	mono	181min	47kbps	119SU	136
0	mono	258min	33kbps	169SU	96

bit3：Reserved

bit2：データ区分　0：オーディオ　1：その他

bit1：再生SKIP　0：通常再生　1：SKIP

bit0：エンファシス　0：OFF　1：ON(50／15μS)

【図２４】

bit7：コピー許可　0：コピー禁止　1：コピー可

bit6：世代　0：オリジナル　1：第1世代以上

HCMS　bit5－4：高速デジタルコピーに関するコピー制御

00：コピー禁止　01：コピー第1世代　10：コピー可

コピー第1世代のコピーした子供はコピー禁止とする

bit3－2：MagicGate認証レベル

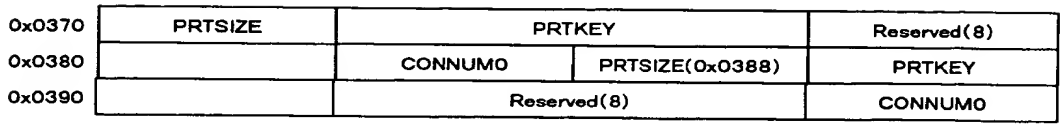
00：Level10(Non－MG)　01：Level1

02：Level2　11：Reserved

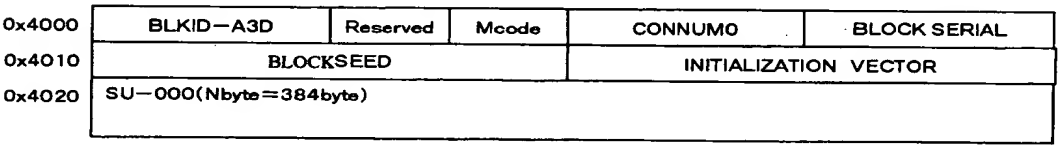
Level10以外はデバイド、コンバインできません

bit1，0：Reserved

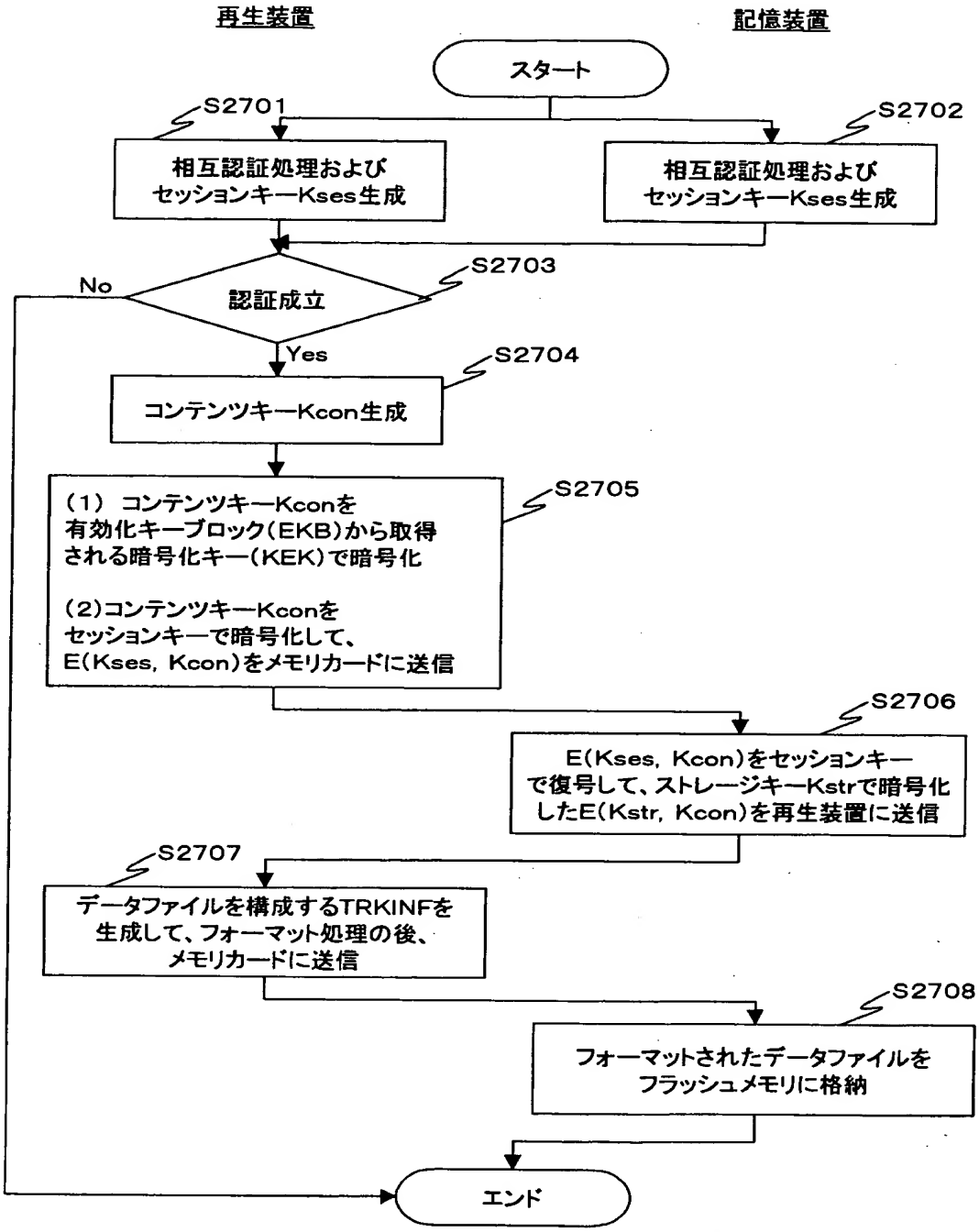
【図２５】


0x0370
0x0380
0x0390
PRTSIZE
PRTKEY
Reserved(8)
CONNUM0
PRTSIZE(0x0388)
PRTKEY
Reserved(8)
CONNUM0

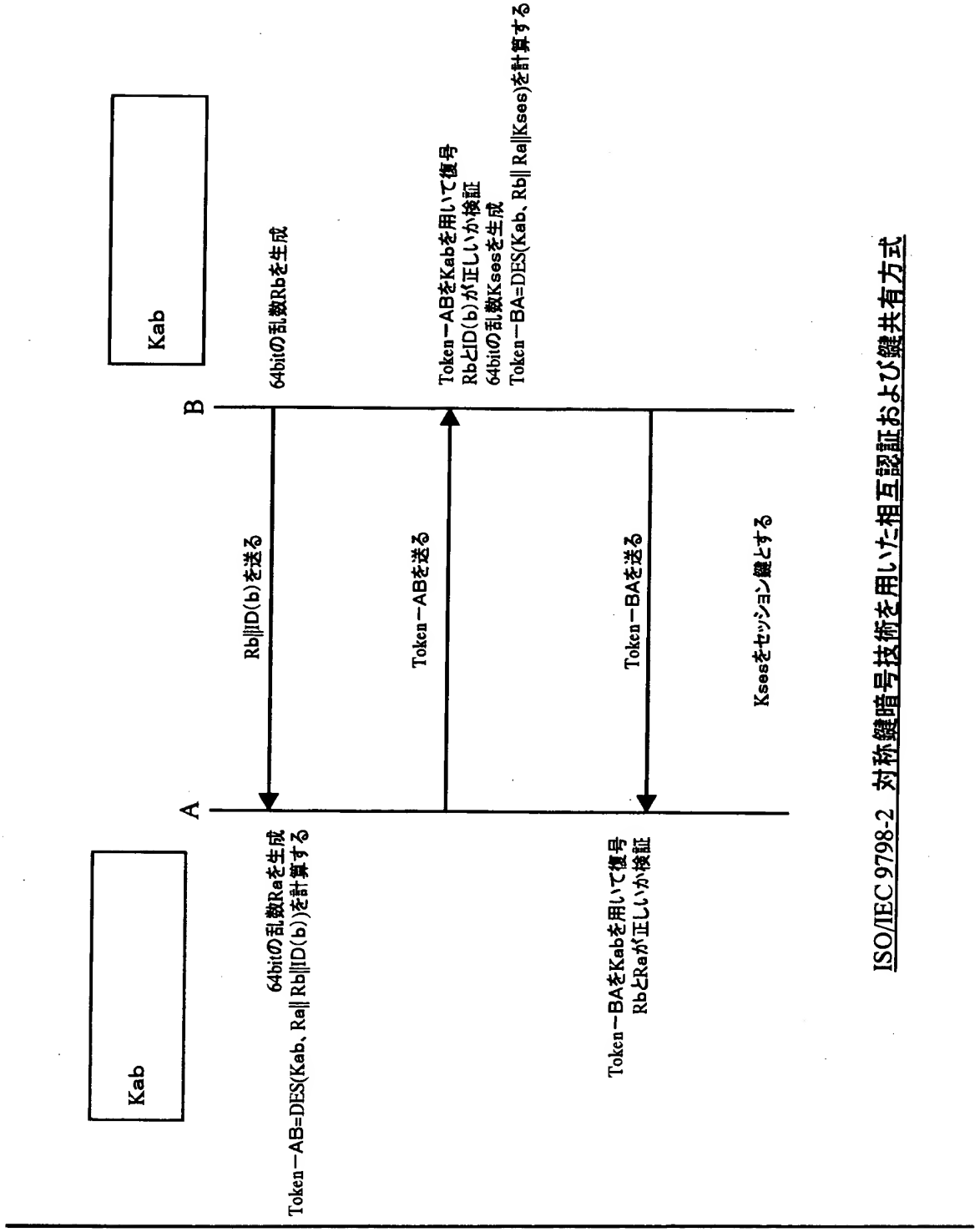
【図２６】


0x4000
0x4010
0x4020
BLKID－A3D
Reserved
Mcode
CONNUM0
BLOCK SERIAL
BLOCKSEED
INITIALIZATION VECTOR
SU－000(Nbyte＝384byte)

【図２７】

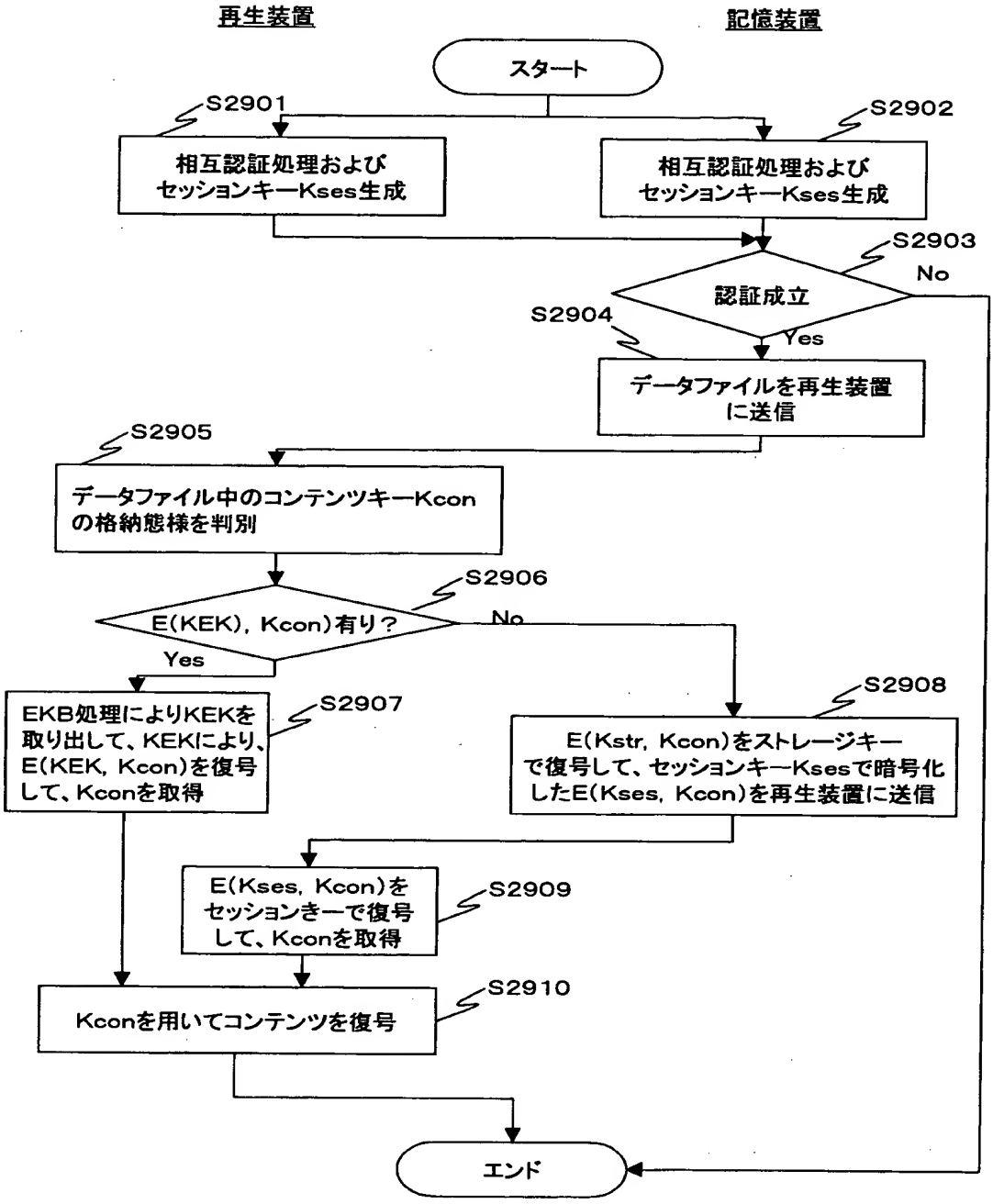

再生装置
記憶装置
スタート
S2701
S2702
相互認証処理および
セッションキーKses生成
相互認証処理および
セッションキーKses生成
S2703
No
認証成立
Yes
S2704
コンテンツキーKcon生成
S2705
(1) コンテンツキーKconを
有効化キーブロック(EKB)から取得
される暗号化キー(KEK)で暗号化
(2)コンテンツキーKconを
セッションキーで暗号化して、
E(Kses, Kcon)をメモリカードに送信
S2706
E(Kses, Kcon)をセッションキー
で復号して、ストレージキーKstrで暗号化
したE(Kstr, Kcon)を再生装置に送信
S2707
データファイルを構成するTRKINFを
生成して、フォーマット処理の後、
メモリカードに送信
S2708
フォーマットされたデータファイルを
フラッシュメモリに格納
エンド

【図２８】

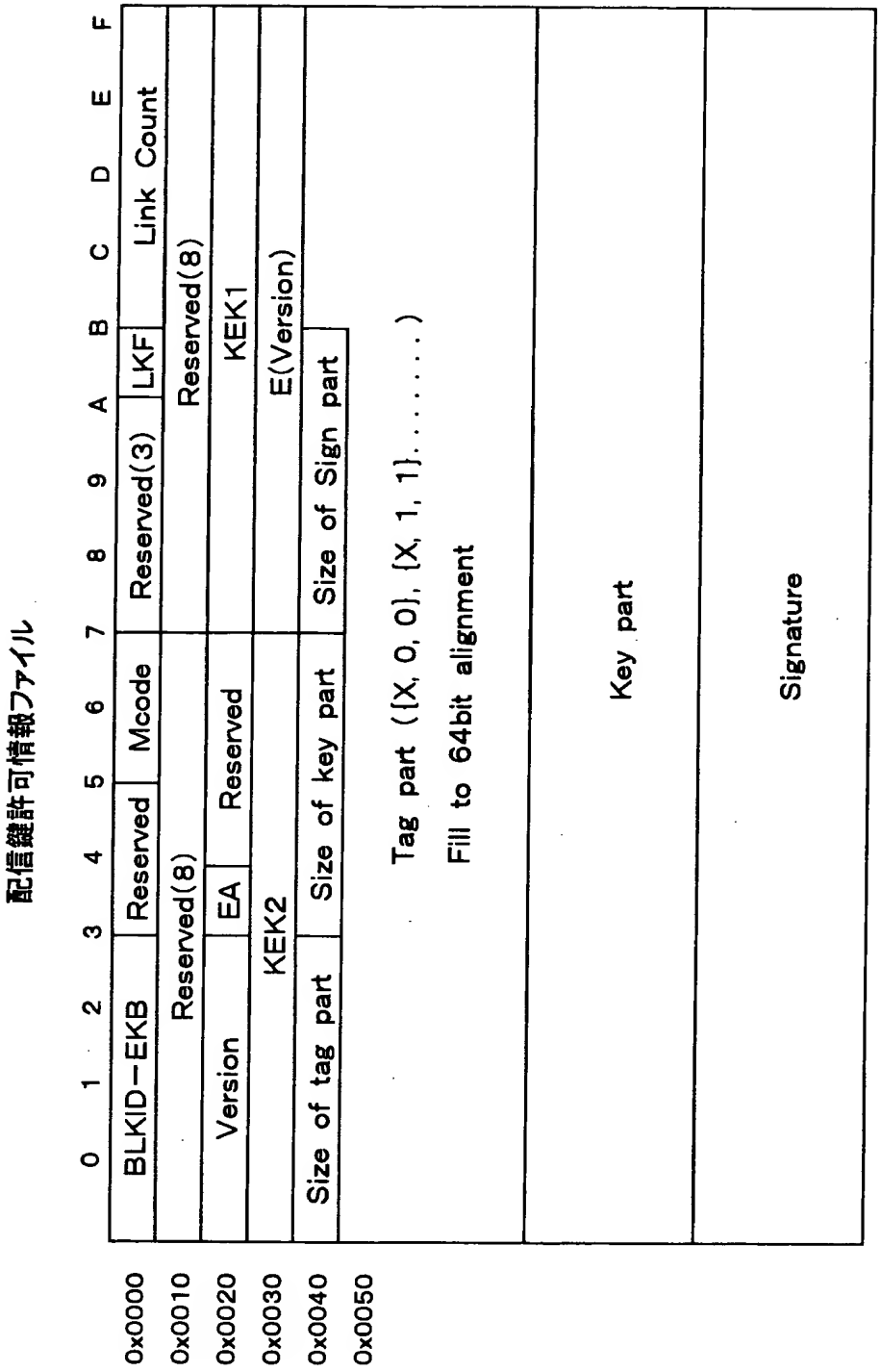


ISO/IEC 9798-2　対称鍵暗号技術を用いた相互認証および鍵共有方式

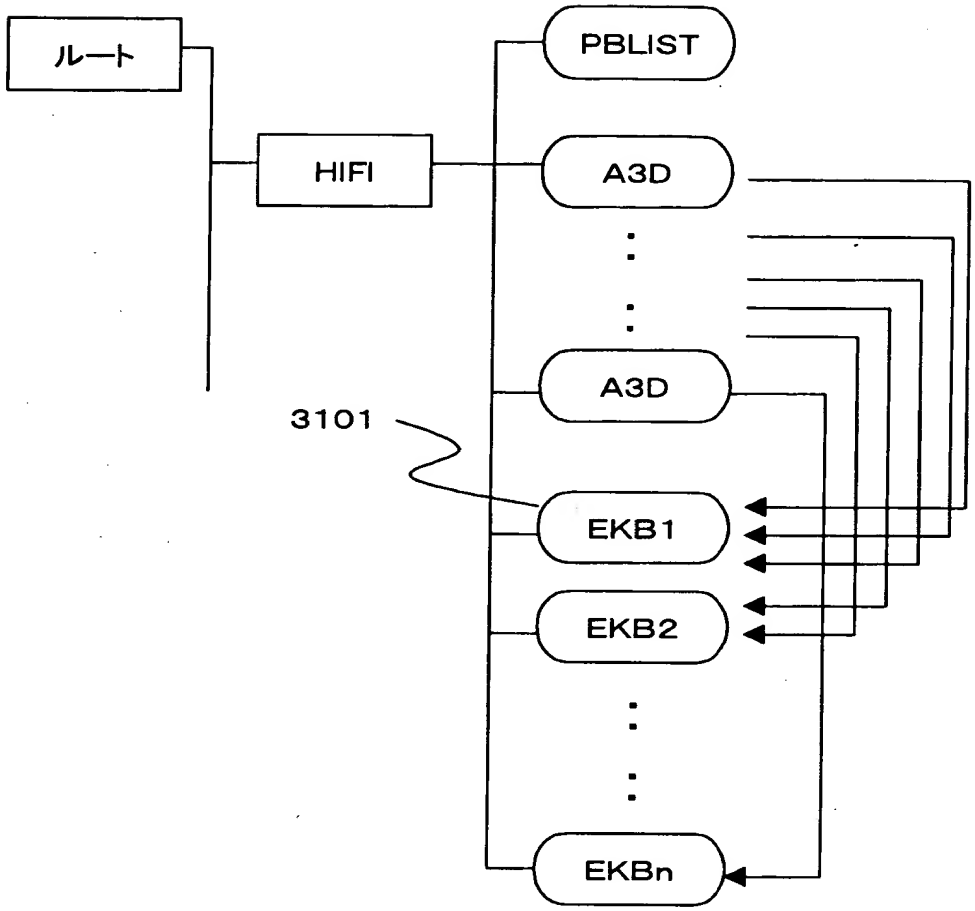
【図２９】


再生装置
記憶装置
スタート
S2901
相互認証処理および
セッションキーKses生成
S2902
相互認証処理および
セッションキーKses生成
S2903
認証成立
No
Yes
S2904
データファイルを再生装置
に送信
S2905
データファイル中のコンテンツキーKcon
の格納態様を判別
S2906
E(KEK), Kcon)有り？
No
Yes
EKB処理によりKEKを
取り出して、KEKにより、
E(KEK, Kcon)を復号
して、Kconを取得
S2907
S2908
E(Kstr, Kcon)をストレージキー
で復号して、セッションキーKsesで暗号化
したE(Kses, Kcon)を再生装置に送信
E(Kses, Kcon)を
セッションきーで復号
して、Kconを取得
S2909
Kconを用いてコンテンツを復号
S2910
エンド

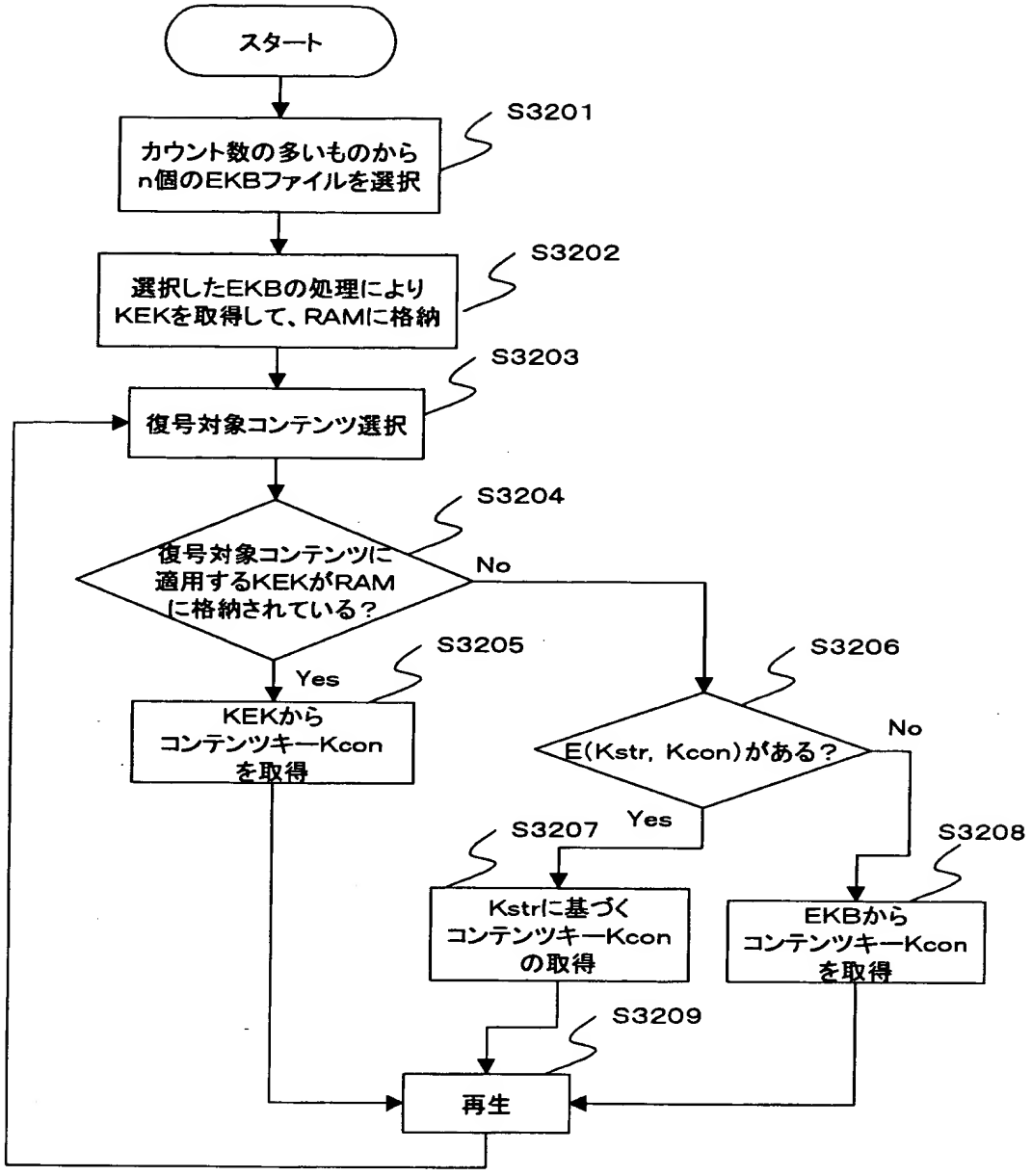
【図３０】


配信鍵許可情報ファイル
0 1 2 3 4 5 6 7 8 9 A B C D E F
0x0000
0x0010
0x0020
0x0030
0x0040
0x0050
BLKID－EKB
Reserved
Mcode
Reserved(3)
LKF
Link Count
Reserved(8)
Version
EA
Reserved
KEK1
KEK2
E(Version)
Size of tag part
Size of key part
Size of Sign part
Tag part ({X, 0, 0}, {X, 1, 1}.....)
Fill to 64bit alignment
Key part
Signature

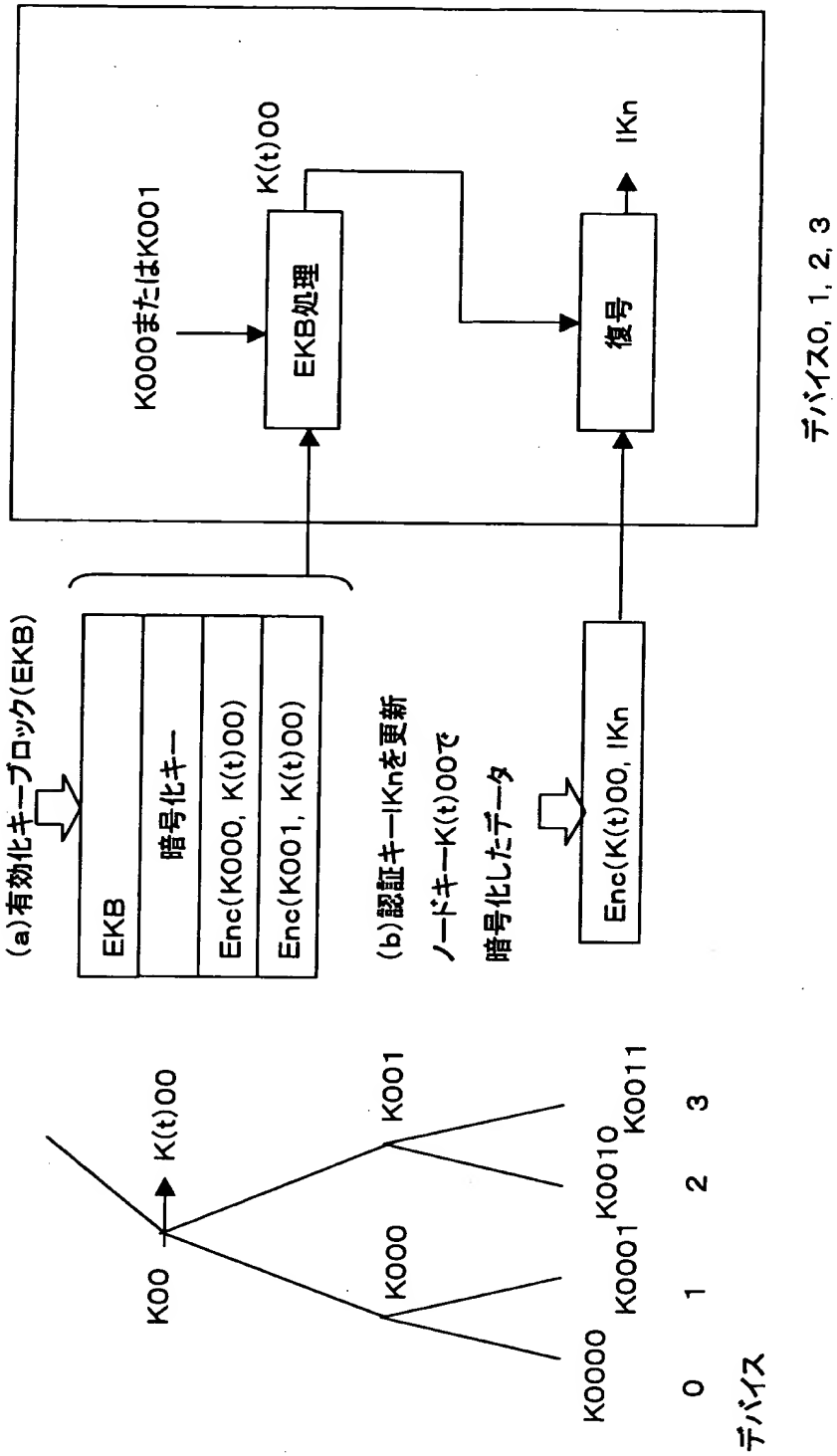
【図３１】



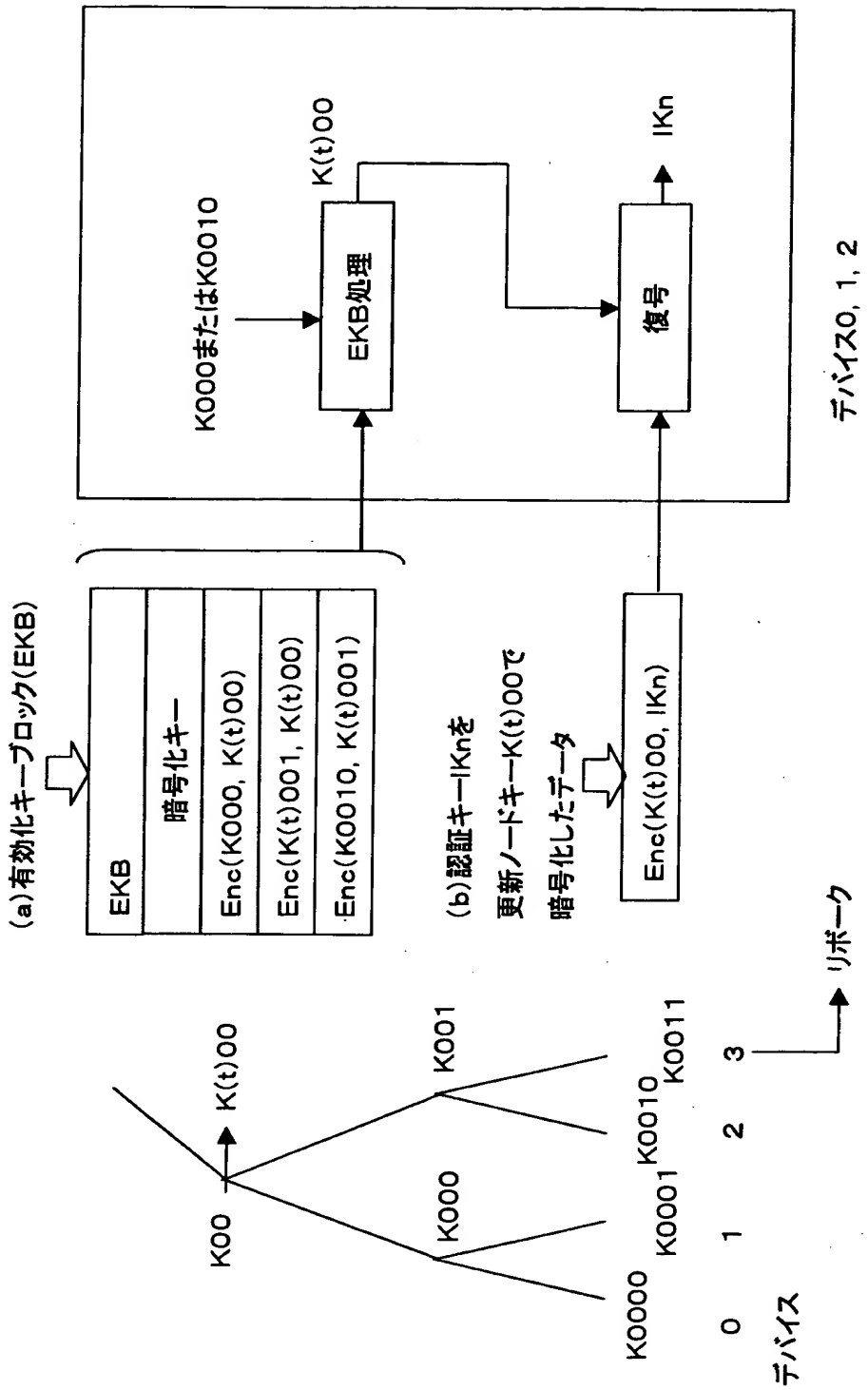
【図３２】


スタート
S3201
カウント数の多いものから
n個のEKBファイルを選択
S3202
選択したEKBの処理により
KEKを取得して、RAMに格納
S3203
復号対象コンテンツ選択
S3204
復号対象コンテンツに
適用するKEKがRAM
に格納されている？
No
Yes
S3205
KEKから
コンテンツキーKcon
を取得
S3206
E(Kstr, Kcon)がある？
No
Yes
S3207
Kstrに基づく
コンテンツキーKcon
の取得
S3208
EKBから
コンテンツキーKcon
を取得
S3209
再生

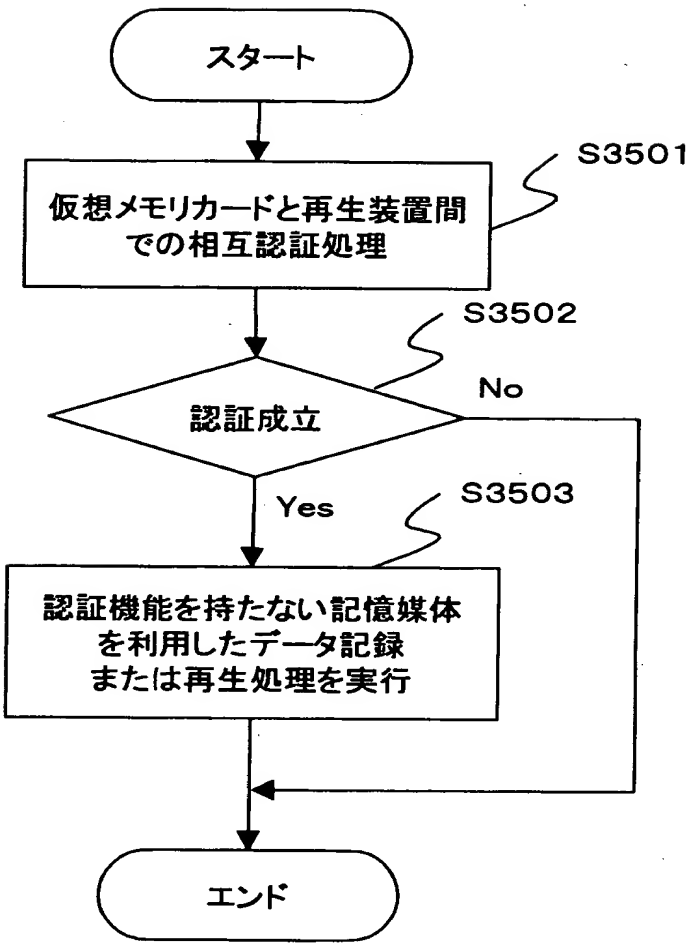
【図３３】


K000またはK001
K(t)00
EKB処理
復号
IKn
デバイス0, 1, 2, 3
(a)有効化キーブロック(EKB)
EKB
暗号化キー
Enc(K000, K(t)00)
Enc(K001, K(t)00)
(b)認証キーIKnを更新
ノードキーK(t)00で
暗号化したデータ
Enc(K(t)00, IKn
K00
K(t)00
K001
K000
K0000
K0001
K0010
K0011
0
1
2
3
デバイス

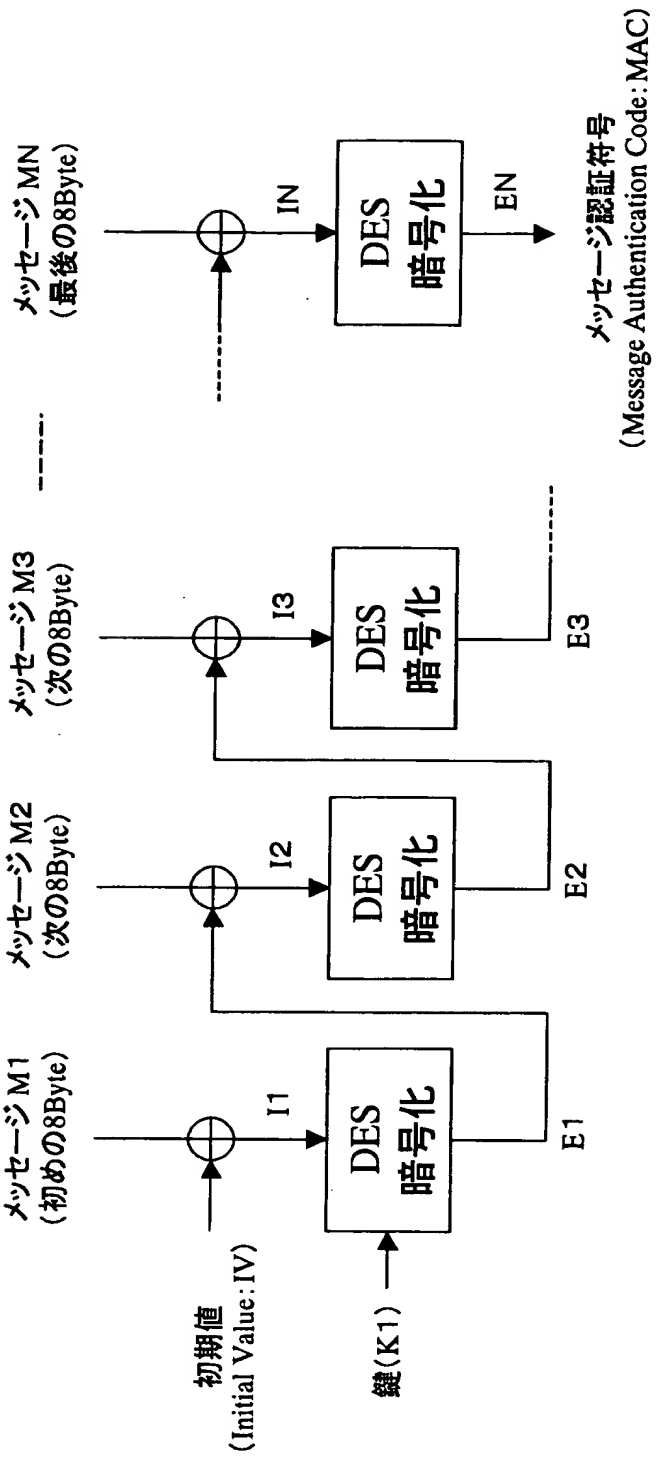
【図３４】


デバイス0, 1, 2
K000またはK0010
EKB処理
K(t)00
復号
IKn
(a)有効化キーブロック(EKB)
EKB
暗号化キー
Enc(K000, K(t)00)
Enc(K(t)001, K(t)00)
Enc(K0010, K(t)001)
(b)認証キーIKnを
更新ノードキーK(t)00で
暗号化したデータ
Enc(K(t)00, IKn)
リボーク
K00
K(t)00
K001
K000
K0011
K0010
K0001
K0000
0
1
2
3
デバイス

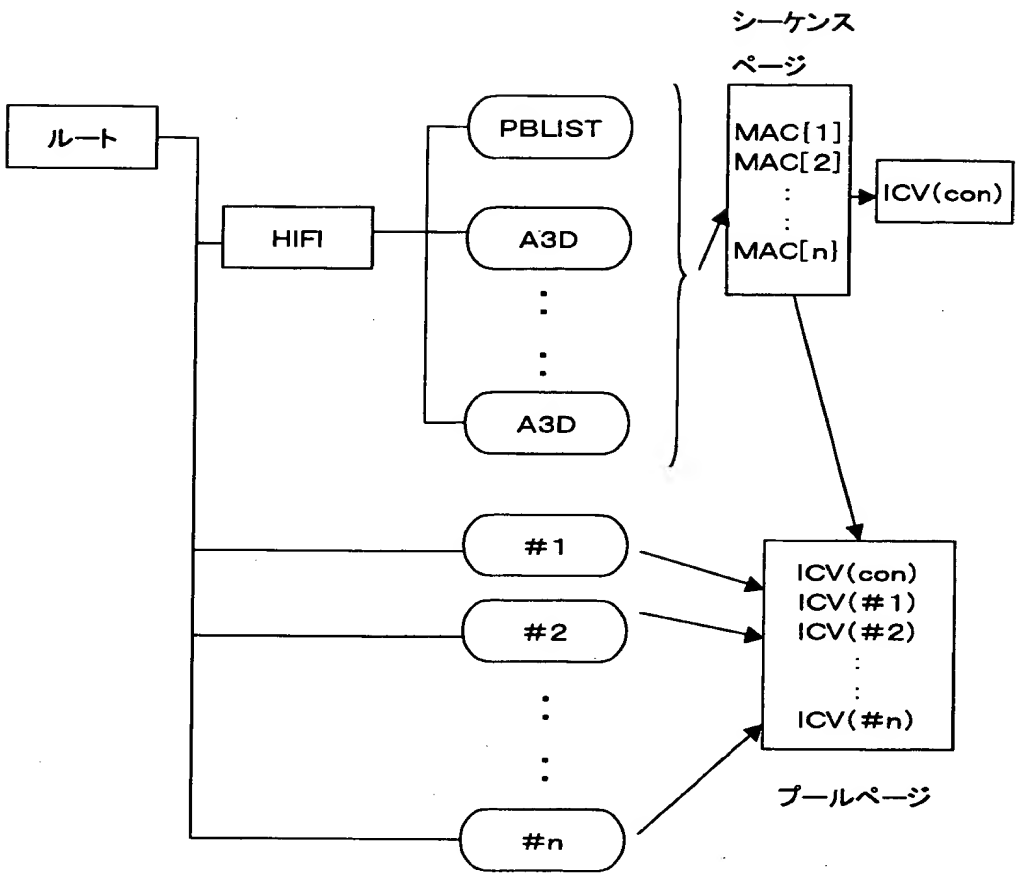
【図３５】


スタート
S3501
仮想メモリカードと再生装置間
での相互認証処理
S3502
認証成立
No
Yes
S3503
認証機能を持たない記憶媒体
を利用したデータ記録
または再生処理を実行
エンド

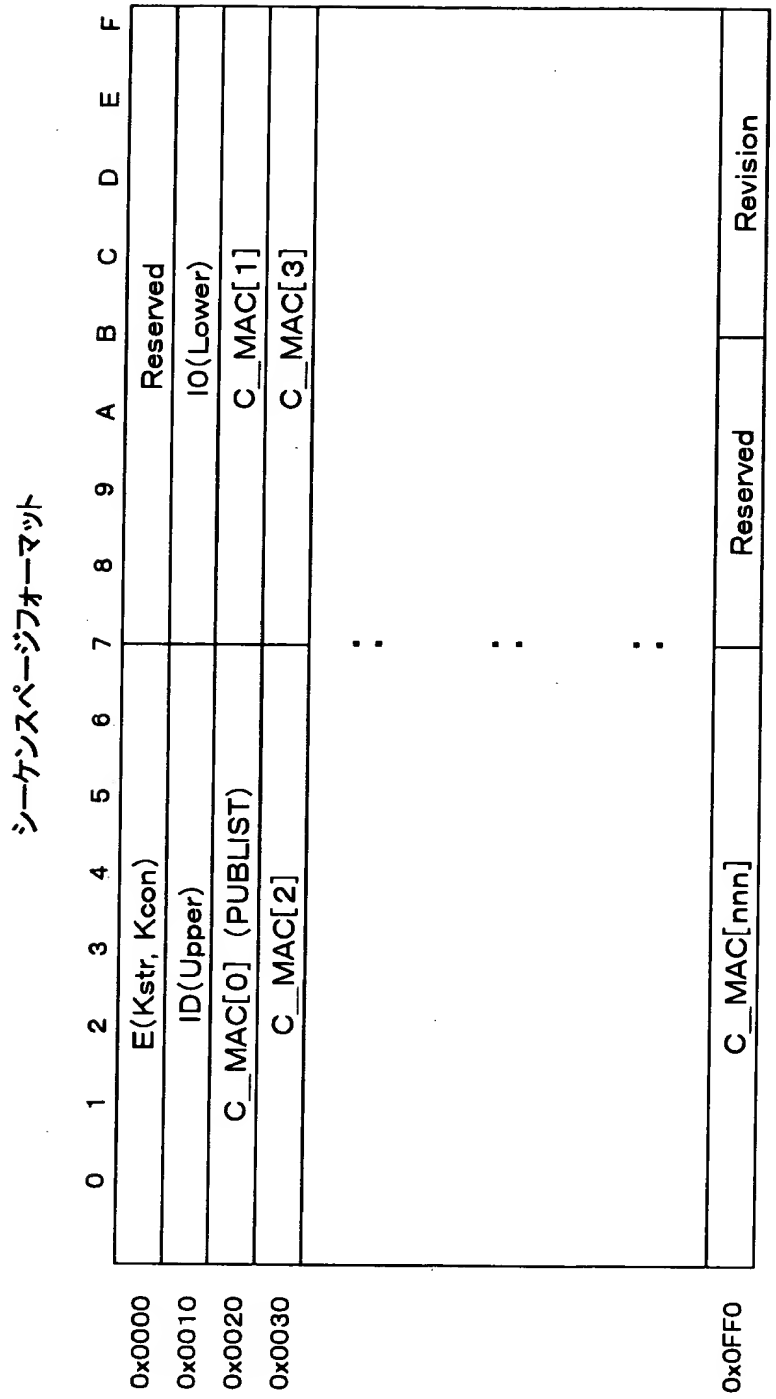
【図３６】


メッセージ M1
(初めの8Byte)
メッセージ M2
(次の8Byte)
メッセージ M3
(次の8Byte)
メッセージ MN
(最後の8Byte)
初期値
(Initial Value: IV)
鍵(K1)
I1
I2
I3
IN
DES
暗号化
E1
E2
E3
EN
メッセージ認証符号
(Message Authentication Code: MAC)
⊕ :排他的論理和処理(8バイト単位)

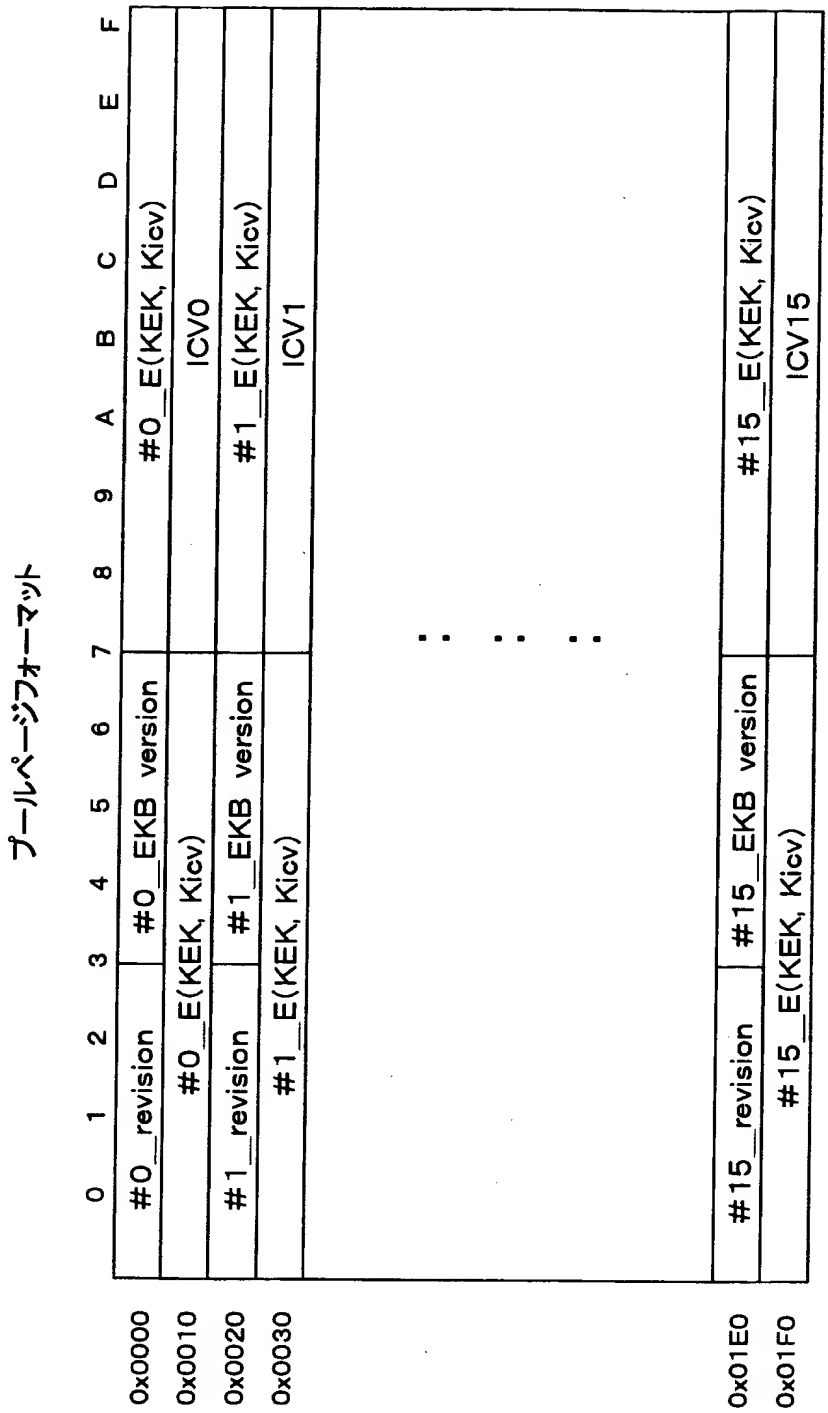
【図３７】


シーケンス
ページ
ルート
PBLIST
HIFI
A3D
A3D
MAC[1]
MAC[2]
MAC[n]
ICV(con)
#1
#2
#n
ICV(con)
ICV(#1)
ICV(#2)
ICV(#n)
プールページ

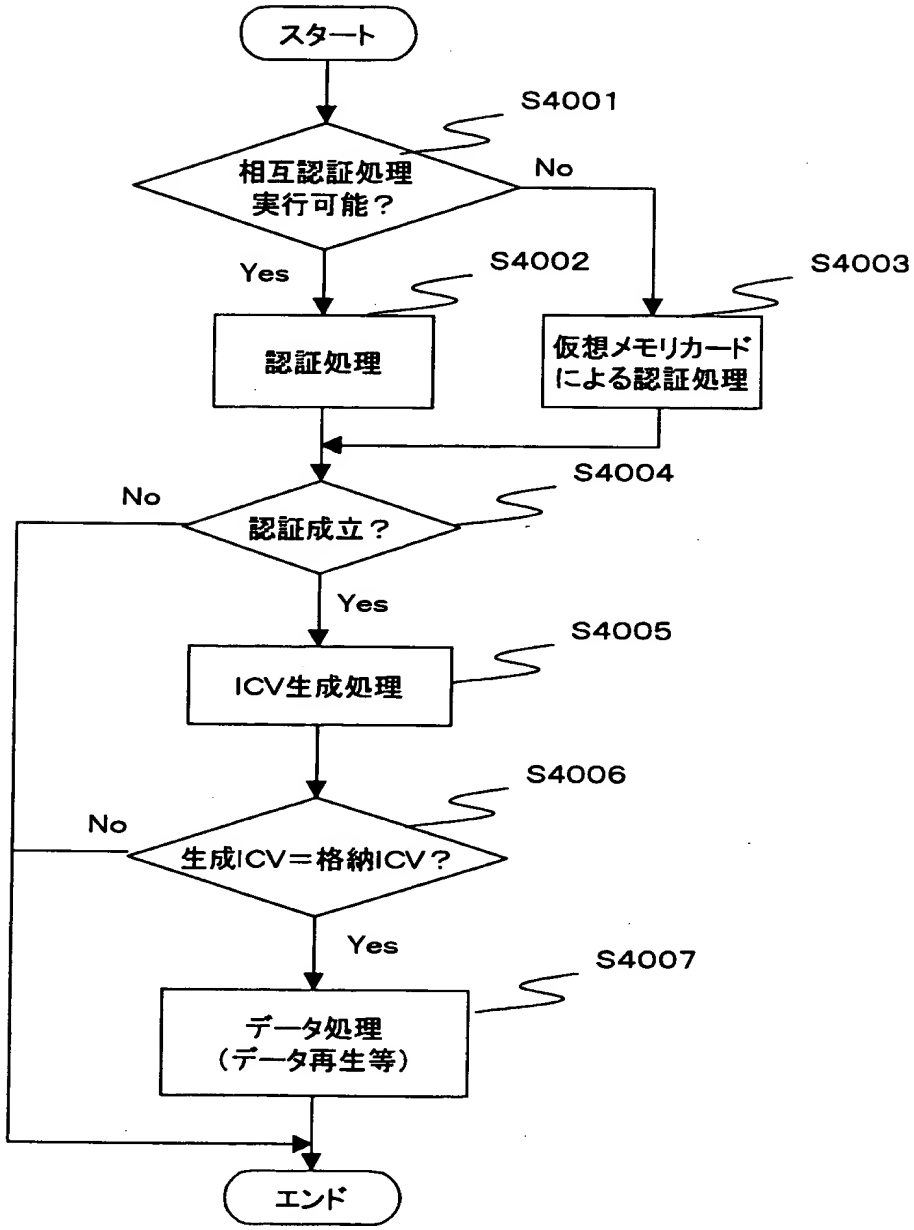
【図３８】


シーケンスページフォーマット
0 1 2 3 4 5 6 7 8 9 A B C D E F
E(Kstr, Kcon)
Reserved
ID(Upper)
IO(Lower)
C_MAC[0] (PUBLIST)
C_MAC[1]
C_MAC[2]
C_MAC[3]
C_MAC[nnn]
Reserved
Revision
0x0000
0x0010
0x0020
0x0030
0x0FF0

【図３９】


プールページフォーマット
0 1 2 3 4 5 6 7 8 9 A B C D E F
0x0000
#0_revision
#0_EKB version
#0_E(KEK, Kicv)
0x0010
#0_E(KEK, Kicv)
ICV0
0x0020
#1_revision
#1_EKB version
#1_E(KEK, Kicv)
0x0030
#1_E(KEK, Kicv)
ICV1
0x01E0
#15_revision
#15_EKB version
#15_E(KEK, Kicv)
0x01F0
#15_E(KEK, Kicv)
ICV15

【図４０】


スタート
S4001
相互認証処理
実行可能？
No
Yes
S4002
S4003
認証処理
仮想メモリカード
による認証処理
S4004
No
認証成立？
Yes
S4005
ICV生成処理
S4006
No
生成ICV＝格納ICV？
Yes
S4007
データ処理
（データ再生等）
エンド

【書類名】　　要約書

【要約】

【課題】　コンテンツの利用を行なうデータ処理装置におけるコンテンツ改竄チェックの効率化を実現した構成を提供する。

【解決手段】　記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行する構成において、検証値をコンテンツのカテゴリ毎に生成して格納する。カテゴリは、コンテンツの種類、あるいは、コンテンツの暗号処理鍵として設定されるコンテンツキーＫｃｏｎを暗号化して提供する有効化キーブロック（ＥＫＢ）の管理エンティティ等に基づいて設定することにより、例えば有効化キーブロック（ＥＫＢ）の管理エンティティ別に、コンテンツデータの検証処理を独立して実行可能となり、効率的な処理が可能となる。

【選択図】　図３７

出　願　人　履　歴　情　報

識別番号　　〔０００００２１８５〕

１．変更年月日　　１９９０年　８月３０日
［変更理由］　新規登録
住　所　東京都品川区北品川６丁目７番３５号
氏　名　ソニー株式会社